ARMY RESEARCH LABORATORY

Efforts in Preparation for Jack Validation

Francisco Azuola
Norman L. Badler
Pei-Hwa Ho
Suejung Huh
Evangelos Kokkevis
Bond-Jay Ting

ARL-CR-418 DECEMBER 1997

prepared by

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, Pennsylvania 19104-6389

under contract

DAMD17 94 J 4486

Approved for public release; distribution is unlimited.

Indigo 2™ is a trademark of Silicon Graphics, Inc.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5425

ARL-CR-418 December 1997

Efforts in Preparation for Jack Validation

prepared by

Francisco E. Azuola
Norman L. Badler
Pei-Hwa Ho
Suejung Huh
Evangelos Kokkevis
Bond-Jay Ting
University of Pennsylvania

under contract

DAMD17 94 J 4486

Blank Pages

Abstract

This document presents a detailed record of the methodologies, assumptions, limitations, and references used in creating the human figure model in Jack, a program that displays and manipulates articulated geometric figures. This report reflects current efforts to develop and refine Jack software to enable its validation and verification as a tool for performing human engineering analysis. These efforts include human figure model improvements, statistical anthropometric data processing methods, enhanced human figure model construction and measuring methods, and automated accomodation analysis.

This report discusses basic details of building human models, model anthropometry, scaling, Jack anthropometry-based human models, statistical data processing, figure generation tools, anthropometric errors, inverse dynamics, smooth skin implementation, guidelines used in estimating landmark locations on the model, and recommendations for validating and verifying the Jack human figure model.

CONTENTS

FIGURES

EFFORTS IN PREPARATION FOR JACK VALIDATION

Chapter 1

Introduction

This document presents a detailed record of the methodologies, assumptions, limitations and references used in creating the human figure model in "Jack," a program that displays and manipulates articulated geometric figures. The model itself is a work in progress, and many issues regarding its performance will continue to improve. Users should be aware that the authors expect constant changes in the human model as more data become available, more research comes to fruition, and more powerful hardware are used. This document will continue to be updated to reflect those changes.

The purpose of having a three-dimensional articulated human model in a graphical environment is to be able to simulate human tasks and functions in a virtual environment before a real-world mock-up is built. In order to fulfill such goals, the human model must be built so that it reflects specific characteristics of real humans. For example, the model must have joint centers, range of motions, and body dimensions comparable to those of its real-world counterpart. This is the goal for the development of Jack human models.

THE HUMAN MODEL

The human model in Jack has gone through many changes over the years, from the skinny body (Grosso et al., 1989) with body segments represented by tetrahedrons of various sizes, to the current body (under construction) with realistic (more human-like) geometric shapes. As more details are added to the model, more data are needed to support this effort.

The human model is composed of segments connected by joints. A segment usually represents a distinguishable part of the body, such as the head. The segment's geometries represent the model's shape. For example, the geometry of the head is associated with the head segment. The articulation is formed by defining joints between segments. For example, the elbow joint is the articulation between the upper arm and lower arm segments. Each joint has an associated range of motion and a number of degrees of freedom (DOF).

Jack uses the polygonal body or "peabody" language syntax to describe figures (usually segmented, but not necessarily human) that are to be displayed or manipulated. A peabody description file allows for the definition of joints, constraints, and sites (attachment points) for

the figures. The human figure model topological structure is defined using a file, referred to in the case of human figures as the "human figure definition file." This file contains other information besides the segment dimensions (geometric surface scale factors). For example, one finds the following information for segments:

- segment geometric surface and its associated scaling factors
- color attributes
- segment mass
- segment center of mass
- segment sites (location and orientation)

For joints, information includes

- connect "site A" to "site B"
- type (degrees of freedom and orientation)
- lower and upper limits for each degree of freedom
- displacements (if any)

ANTHROPOMETRY-BASED HUMAN MODEL CONSTRUCTION

This section addresses assumptions and limitations in building the model, which are mostly attributable to insufficient data and the lack of an "absolute" analytical or mathematic model.

Joint Center Locations

The location of joint centers, which collectively define the model link structure, is estimated based on a biostereometric human body sample. To date, no anthropometric survey provides joint center location data. Further work is needed to provide quality estimations.

Independent Degrees of Freedom

The range of motion specified in our model came from studies conducted by NASA (1987). Each degree of freedom is measured independently. In contrast, the degrees of freedom for human beings are not independent of each other, either within the same joint or sometimes across joints. This is because of the structure of joints and the muscle groups that control them. Thus, it is possible for the figure model to be positioned in a way that looks awkward or unrealistic.

Floating Anthropometric Measurements

The current data from standard anthropometric surveys are collected with devices such as tape measures, calipers, and anthropometers (Roebuck, 1995). These measurements are numerical values, many of which do not refer to a fixed reference in three-dimensional (3D) space. When such data are used to build 3D human models, the figure model designer must decide where and how they should be employed.

Model Geometries and Dimensions

An anthropometry-based human model needs to have built-in support for scaling so that models of different dimensions can be created. The various 3D shapes of the human body segments require a more detailed description than can be provided by single dimensional anthropometric data. Thus, when scaling is applied to change the model's dimensions, certain assumptions have to be made about the geometries to compensate for insufficient data. An example of these assumptions is the adoption of generally standard shapes to model segments, thus reducing the number of parameters needed to control the geometry's size and shape. The polygonal body model uses linear, bounding box[1] scaling for all geometries (except arms and fingers) in which tapering scaling is used. A smooth skinned body is now available, where the geometries remain continuous across joints when joint angles are changed. It uses realistically shaped geometries to better represent the human in 3D. More sophisticated scaling methods are being developed that would provide higher levels of control of body shapes and could be used to size the body to different dimensions.

RECENT DEVELOPMENTS

This section outlines major improvements and work related to the human model done within the past year.

Current efforts include statistical anthropometric data processing methods, enhanced human figure model construction and measuring methods, and automated accommodation analysis.

Cervical Joints

The Jack human figure now has cervical vertebrae. The cervical joints can move in a manner similar to the movement of the 17-segment spine. There is a set of default degrees of

[1] a rectangular, graphical scaling aid

freedom and ranges of motions. The user is allowed to redefine them or to "freeze out" certain portions of the cervical column in order to model more restrictive neck movements in the presence of constraints or injuries. Adding cervical joints has greatly enhanced head movements. The head can now move more realistically because of the more refined distribution of neck bending, twisting, and lateral bending.

Spine Curvature Improvements

The spine of the previous model was barely curved. This contradicts what one finds in a real torso. The current model has a curvature consistent with that of an average human. The flexible torso range of motion should be more realistic because of improved locations of joint centers along the spine.

Visible Landmarks

Various utilities have been developed to define landmarks, place and move landmarks to arbitrary positions, color landmarks, label landmarks, and display landmark labels.

These utilities were developed with the understanding that, as the human model improves in its representation of a real human body in terms of link structure and geometric segment shapes, anthropometric landmarks will play a crucial role in defining various properties of the model.

Automated Accommodation Analysis

An automated accommodation testing utility has been developed that includes any posture defined by the user. The utility takes a user-defined environment, which may contain multiple human figures, and tests all human figures in a user-specified directory. Each figure is placed in the same environment under the same human behavior control and constraints. The utility then evaluates all constraints and records the offset distances for them. Currently, the utility can test as many as 500 figures at a time, at a rate of about 10 seconds per figure on an Indigo 2™ workstation with 64 megabytes of memory. The results of the testing are written to a plain American standard code for information interchange (ASCII) file for post-processing and analysis.

The utility is very useful when evaluating a work space design. Depending on the requirements of the designer, a work space can be evaluated with a few, a hundred, or even thousands of figure models. Combined with the proper figure generation algorithms, this utility

automates part of the human engineer's simulation tasks and provides useful quantitative data for detailed analysis.

A Monte Carlo-based figure generator (randomly selecting anthropometric parameters) has been developed to supply the test simulation with the necessary test cases. In addition, a complementary cadre or boundary family generator (representing extremes of population characteristics) has been developed.

Improved Surface Geometry and Smooth Skin

With improvements in hardware performance, more realistic and non-rigid geometries for the human model can be used.

A male model is constructed with realistic geometric shapes. Utilities are provided to deform the geometries so that, after a joint angle changes, the geometries around the joint remain smooth. The same technique is applied to the torso, neck, and hip, so that the full body is smooth (except for the hands and fingers).

Dynamics and Joint Torque Loading

Jack now has its own dynamics computation modules, which allow for the computation of joint torques when a load is attached to the human figure. This feature gives the user the ability to compute such loads interactively. It also facilitates the comparison of such computations with experimentally obtained data.

Anthropometric Methods

A figure model has been created following an object-oriented approach. The figure object consists of the data storage structures plus two main methods, namely, a constructor and a measuring method. All the rules used by both methods are redefinable, with defaults provided. Having the construction and measuring rules exposed simplifies the task of validating, verifying, improving, and updating the model. The figure object's implementation has been highly optimized for efficient creation of large populations of figure instances.

CONCLUSIONS

To evaluate any anthropometry-based human model, one must establish a suite of tests and measurements that can quantify a model's performance with respect to clearly defined tasks

and goals. From a user's point of view, such tests can be used to select a model best suited for specific tasks. From a modeler's point of view, they can identify limitations of a model and perhaps suggest improvements.

The most basic tests should include anthropometric measurements of the model against the survey from which the model has been constructed. Of the hundreds of traditional anthropometric measurements, a subset should be selected that focuses on the part of the human model most relevant to the intended use or tasks.

The next level of tests should focus on the model's functional performance or task-specific analysis. Possible candidates for these tests are functional reaches and reach space for a specific joint chain. These tests could reveal the model's validity in the placement of joint centers, joint limits, distribution of joint displacement in different postures, and control of multiple joints.

Another test should be designed to test the dynamic capabilities of the models. Such a test should compare the computed forces and torques in the model against experimentally obtained data.

The interpretation and analysis of test results should be based on multiple tests on various sized models generated from the same modeling system, with specific figure generation schemes (e.g., Monte Carlo). Statistics should be compiled for each modeling system so that meaningful, task-specific or posture-specific comparisons can be made.

OTHER IMPORTANT REFERENCES

To trace the Jack human model developments and evolutions, interested parties can refer to the following documents for detailed information:

- Anthropometry for Computer Graphics Human Figures (Grosso et al., 1989)
- 1988 Anthropometric Survey of U.S. Army Personnel: Methods and Summary Statistics (Gordon et al., 1989)
- SASS version 2.5 User's Manual (Azuola, 1995)
- Jack 5 User's Guide (Phillips, 1991b)

A web site can also be checked for the latest announcements:

http://www.cis.upenn.edu/~hms/jack.html

Chapter 2

Virtual Human Models

This chapter describes the construction of virtual human analogs that model the structure of real humans. Such analogs are used to simulate human functions in a virtual environment.

We start by examining human body structure, followed by a review of issues in building such models.

THE HUMAN BODY

Anatomically, the human body is composed of skeletal system, muscular system, inner organs, skin and fat layers (Basmajian, 1982; Vander, Sherman, & Luciano, 1980). The skeletal system and muscular system, jointly called the musculoskeletal system, protect, support, and move the body, while the organs maintain a stable internal environment (homeostasis). The overall shape of the human body is determined mostly by the amount of fat and muscle it has and how they are distributed. The size of the body is determined by the skeleton and the soft tissues that cover it.

The human skeletal system is composed of more than 200 bones. Bones are connected by joints. Collectively, they form the articular system, which determines the maximum amount of flexibility, mobility, and range of movements for the body. Such limits are affected by many other factors such as the amount of fat, clothing, equipment attached to the body, age, or physical health. The skeleton's size, structure, and proportions are affected by age, exercise, genetics, nutrition, and even profession (Roche, 1978), but interest is focused on modeling the skeleton of an average, healthy adult.

Joints in the human body are very complicated. Some types of joints, such as the fibrous and cartilaginous joints, allow little, if any, movements. Others, such as the synovial joints, allow specific types of movements. Synovial joints are the most important joints for human posture and motions (Basmajian, 1982). They can be classified into uni-axial (hinge), bi-axial (condyloid), and multi-axial (plane, ball-and-socket, saddle, pivot) joints (Basmajian, 1982).

Muscles provide the needed forces for the body to support itself, move, and posture. Muscles often work in groups. For example, the elbow is controlled by the biceps and triceps. (Basmajian, 1982). The correspondence between joint and muscles is not only one to one or one

to two but can be one to many and many to one. The spine, for example, has 34 vertebrae, 35 joints, and more than 20 muscle groups to support its movements and postures (Basmajian, 1982; Monheit & Badler, 1991). The shoulder is another example, with multiple muscles jointly controlling multiple joints (Zhao, 1993; Otani, 1989; van der Helm, Veeger, Pronk, van der Woude, & Rozendal, 1991).

Over the skeleton and muscles, there are soft tissues and skin. The latter form the final shape of the body. The amount of muscles and fat tissue are the dominant factors in shaping the body. These factors, in turn, are affected by things such as genetics, exercise, and dietary habit (Bailey, Malina, & Rasmussen, 1978).

VIRTUAL HUMAN MODELS

A virtual human models only a subset of the anatomical and biomechanical aspects of a real human, depending on the purpose and level of abstraction. The main focus is on the construction of a scalable model that represents an average human in size and possesses similar flexibility and range of movements. This requires having realistic geometric shapes, an acceptable representation of the articular system, and proper sizing algorithms.

Joint and Joint Centers

For the model to behave properly, joint locations should be specified appropriately within the human body segments. Also, it is necessary to specify the way they rotate or translate about the correct axes with proper limits. When the geometries and links are sized to different dimensions, it is important that the characteristics of the articular system are passed along to the new (size) figure instance (a figure with a particular set of parameters). Figure 1 shows one of the polygonal human models, polybody (Phillips, 1991b).

If all the reachable points for the end effector of a single joint are lumped together as a reach space (Alameldin, 1991), then an incorrectly placed joint center or joint axis will cause a shift or rotation of such reach space relative to the correct joint center, while incorrect joint limits will enlarge or reduce such reach space (Roebuck, 1995). When more than one joint is involved, errors in modeling joints affect the size, shape, location, and orientation of reach space.

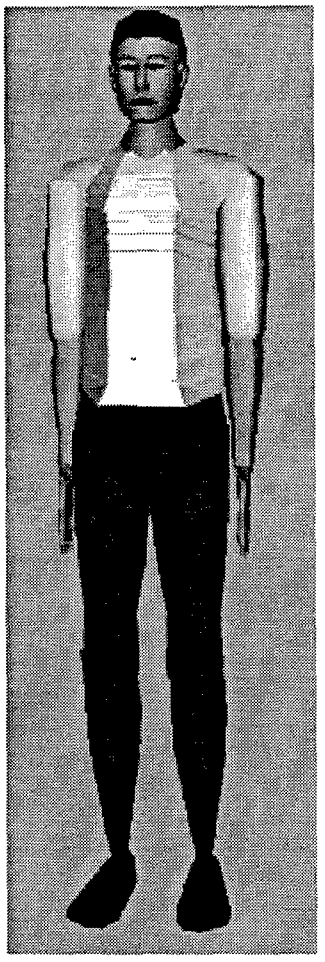

Figure 1. Polybody human figure model.

Techniques developed in the field of robotics (Paul, 1981; Yoshikawa, 1990) are often used for modeling and controlling joints in virtual human or character models. When "robot joints" are used to model joints in a human model, their articulation can be uniquely defined by their link mechanisms. These mechanisms state the linkages (joints) among the segments (links). Linkages are specified within each segment's local coordinate frame (Phillips, 1991c; Paul, 1981). To go from one segment to its neighbors, one simply performs the geometric transformations associated with the sites that define the joint.

Since the joints in the human body usually do not have perfect spherical shape and seldom rotate around a fixed point or plane, additional errors are introduced if "robot joints" are used. This is an area where better joint models require more computation and thus, adversely affect system performance. A decision has to be made between better performance or better accuracy. Sometimes, errors from other sources may be much more serious than modeling errors. It follows, then, that "robot joints" prove to be a reasonable choice (Zhao, 1993).

Model Geometry

Geometries for a human model can be obtained in several ways. Artistically, one creates the geometries that may or may not resemble a real human. The level of detail and realism achieved depends on the skills of the modeler (artist). The polygonal human model in Figure 1 is composed of about 2400 simple polygons. It has a rather artificial appearance. Geometric shapes can also be obtained by scanning real human subjects with the help of imaging or photographic equipment (Chen & Zeltzer, 1992). The level of detail can be extremely high, yielding very realistic representations. Figure 2 is the human model from Viewpoint Datalabs, created by medical illustrators. It has close to 40,000 polygons. In general, human body geometric shapes are available in all levels of detail.

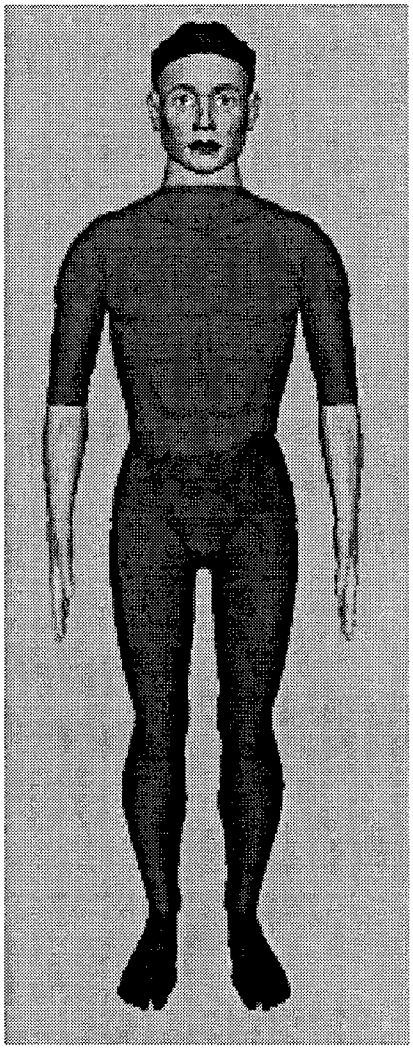

Figure 2. Viewpoint human figure model.

Depending on the source of the geometry, we may or may not know all the associated physical and physiological attributes. Attributes related to the geometry itself, such as area, distance, or volume, can mostly be estimated numerically. Those related to the physiological properties of human bodies can only be obtained from real human data.

Control and Manipulation

Once a human model is constructed, it can be used for a wide range of interests, from entertainment to simulation. The intended purpose of our research is to provide realistic 3D human models for human factor engineers, to facilitate their evaluation of work space designs. For this purpose, utilities need to be developed so that the human model can be easily put into various postures where reach, field of view, and space accommodation analyses can be conducted (Badler, Phillips, & Webber, 1993; Phillips, 1991a). Without realistic human models, however, little value can be generated with such tools. It is our goal to generate figure instances that are realistic, not only in shape, but in their anatomical, biomechanical, and anthropometrical configurations, so that these figure instances prove useful when applied in human factor analyses.

Chapter 3

Human Figure Model Anthropometry

In evaluating a work space design, one needs not just one human model, but many models of different sizes. To have models of various sizes means that there are (a) mechanisms to control and measure the size of the model, and (b) established sources of human data to support the sizing operations. For the latter case, one finds many anthropometric surveys providing data sets about populations of human subjects. We address measurements of the model in this chapter. Chapter 4 addresses the issues of scaling the human model, and Chapter 5 discusses how the 1988 Anthropometric Survey of U.S. Army Personnel (ANSUR 88) is used to generate the Jack figure models.

MEASUREMENTS OF THE HUMAN MODEL

Human models can only be specified by parameters or attributes that can be computed (directly or indirectly) on the model. For instance, segment length can be used if one can compute segment length on the model. As another example, density cannot be computed from the model's associated geometries directly but can be associated with the model so that properties such as mass can be computed. Given volume and mass, density can be derived or computed indirectly. Hence, the density parameter can be used and changed in a model, even though no physical or visual changes can be associated with such a parameter.

Distance

Distance is an attribute that can be computed exactly between two 3D points in virtual space. It is associated with attributes such as length, width, and depth.

Circumference

Circumference cannot be computed exactly in models, since the geometries used are an approximate representation of the real human geometric shapes. Circumference must be defined relative to a fixed plane or a cross section, but the population surveys do not provide support information. This leads to differences between the real human and the model, thus introducing error.

Volume

Volume is another attribute that can only be viewed as an approximation of real human segment volume. When density is used in a human model, the mass of that model can be computed once volume is obtained. Between volume, density, and mass, there are only two free parameters. The third one can be derived from the other two.

SIGNIFICANCE OF MEASUREMENTS

It is important that measurements of human models are viewed with the limitations of such models in mind. Circumference, for example, is affected by the plane on which it is computed and by the realism of the model geometries. A model can have good distance accuracies if it is a good representation of the human body in terms of link length and joint center locations. Circumference and volume, for example, are more difficult to represent since they require accuracy in more than one dimension.

Chapter 4

Scaling Geometric Objects

In this chapter, we discuss issues in scaling geometric objects. Techniques that can be applied to the Jack's scalable models are presented.

SCALING DEFINED

Scaling is a transformation process that changes a geometry's dimensions, shape, or both. The process is not affine (Farin, 1990), since it does not preserve shape or angles. We differentiate dimensional scaling from shape control. The former emphasizes controlling the dimensions of geometries and the latter focuses on non-quantitative means to control shapes of geometries. The two are somewhat related, since scaling can change both dimensions and shapes.

Scaling can be defined as a function $S(x, y, z)$ applied to a geometry $G(x, y, z)$ to obtain a new geometry $Gnew(x, y, z)$.

$$Gnew(x, y, z) = S(x, y, z) \bullet G(x, y, z) \tag{4.1}$$

DIMENSIONAL SCALING

Dimensional scaling of geometries transforms geometries so that they satisfy specific dimensions. For anthropometry-based human models, the types of specifications include measurements of body segment length, depth, and width.

Depending on the specifications, different scaling approaches may be required. We can start with linear scaling and then move into more advanced scaling techniques to support more flexible specifications.

NORMALIZATION

To prepare a geometry for scaling, a coordinate frame has to be chosen. Often, the geometry is reduced to unit dimensions for subsequent scalings. This process is called normalization. It is equivalent to placing the geometry into a bounding box centered around the origin of the coordinate frame. Once a geometry is normalized, it can be scaled by specifying how the bounding box is stretched by various functions (constant, linear, nonlinear, etc.).

Figure 3 shows a human head model, before and after normalization, with the bounding box shrunk to a 2x2x1 cube.

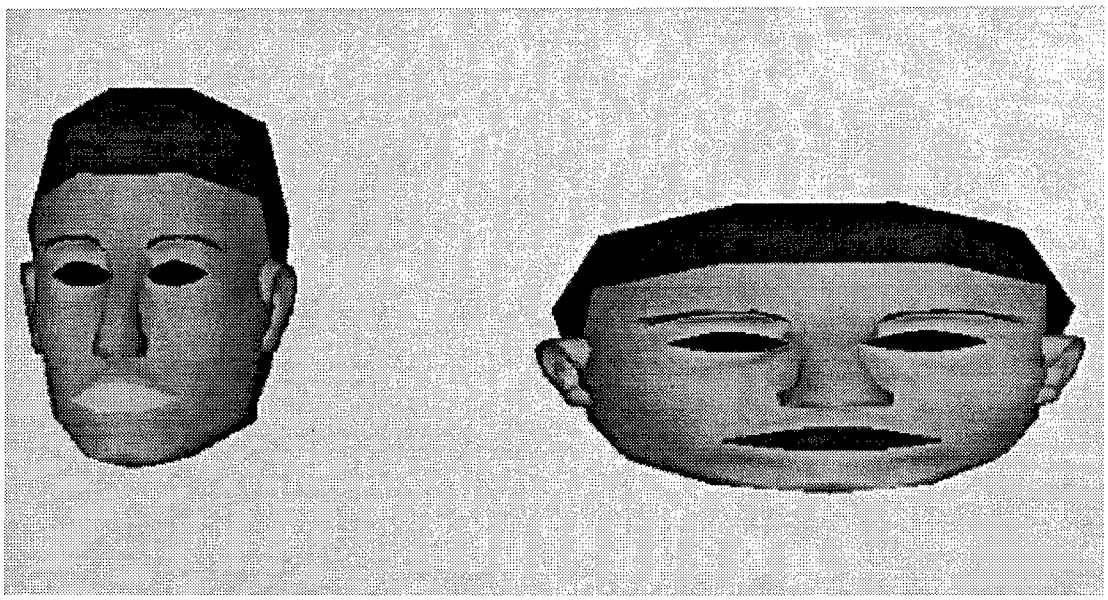

Figure 3. Original and normalized (enlarged 16 times).

LINEAR SCALING

Linear scaling is the simplest type of scaling. A geometry is simply stretched or shrunk uniformly in each dimension. This technique is compatible with normalized geometries. This is also called a bounding box approach, since the process simply scales the bounding box uniformly with the geometry in the box. Linear scaling can be defined by

$$Gnew_x = Sx \cdot G_x$$
$$Gnew_y = Sx \cdot G_y$$
$$Gnew_z = Sx \cdot G_z$$

in which *Sx*, *Sy*, and *Sz* are constants.

With linear scaling, only specifications that are related to the three orthogonal dimensions (i.e., length, width, and thickness) can be used. Circumference can be used as well, but it is related to two dimensions that define the plane where it is taken. Given the circumference value, one is left with an additional degree of freedom for the scaling algorithm.

Provided that the reference frame is established correctly, the bounding box approach is suitable only for symmetric objects such as cylinders and ellipsoids. These objects have their axes of symmetry at the center of the object and coincide with the axes of their coordinate frames. Objects that are not symmetric or irregular, such as human body segments, do not scale well with this approach.

EXTENSIONS OF LINEAR SCALING

One simple extension of linear scaling is what we call tapered scaling, in which different scale factors are assigned to different nodes along the same axis. The underlying geometry gets scaled by the exact scale factors at the nodes. Between nodes, a linearly interpolated scale factor is applied. This technique can be used to scale an object that has different scaling specifications at various points. Forearm and thigh segments are prime candidates for this type of scaling, where the widths and depths at the two ends may need to be scaled differently.

We can define segmented scaling, assuming that the geometry is segmented, into n regions along the z axis by nodes $z_0, z_2 \ldots z_n$ as follows:

$$Gnew_x = \begin{cases} Sx_i \cdot G_x & \text{for } z = z_i \\ Interp_x(z) & \text{for } z_i > z > z_{i-1} \end{cases}$$

$$Gnew_y = \begin{cases} Sy_i \cdot G_y & \text{for } z = z_i \\ Interp_y(z) & \text{for } z_i > z > z_{i-1} \end{cases}$$

$$Gnew_z = G_z$$

for $i = 0, 1, 2 \ldots n$ and *Interp* is a linear interpolation function (Farin, 1990) defined as

$$Interp_x(z) = \frac{z_i - z}{z_i - z_{i-1}} Sx_{i-1} + \frac{z - z_{i-1}}{z_i - z_{i-1}} Sx_i \qquad \text{for } z_i > z > z_{i-1}$$

$Interp_y$ can be defined similarly.

This method provides better shape control compared to the bounding box approach but only works well with objects that are symmetrical at each cross section along the long axis.

NON-UNIFORM SCALING

To overcome the constraints imposed by linear and tapered scalings, non-uniform and nonlinear scalings are used.

Figure 4 shows the original and non-uniform scaled leg of a human model. The scaling function used is a combination of sinusoidal functions, *sine* and *cosine*, and linear interpolation. The scaled geometry has a different shape and continuity is maintained.

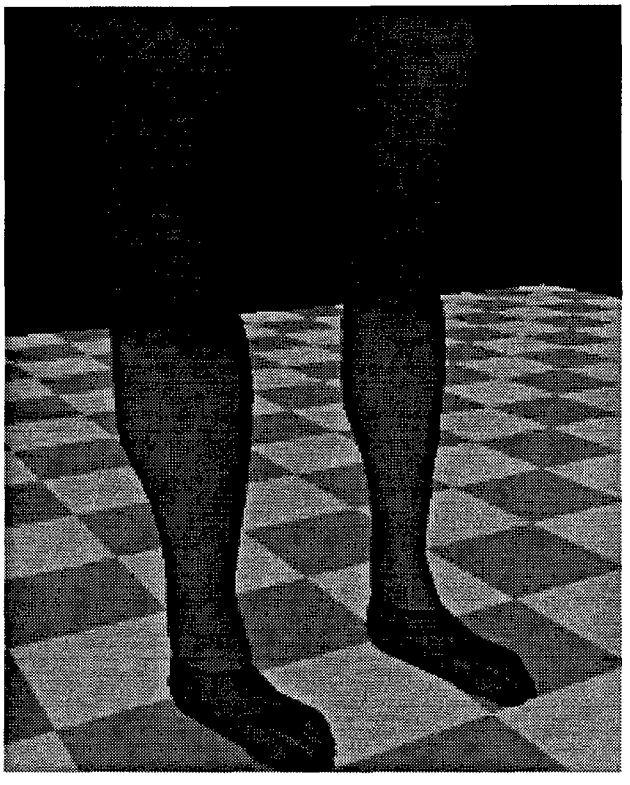

Figure 4. Results of non-uniform scaling.

SUPPORTED SPECIFICATIONS

With linear scaling, length, width, and depths are directly supported. The disadvantage of using those specifications is that resulting geometries are "boxy." Moreover, graphical continuity may not be preserved, since two neighboring geometries may have different specifications. Figure 5 shows a model leg that has discontinuities at the knee because of linear scaling. To ensure continuity, all neighboring geometries should have the same scale factors at their plane of intersection with their neighbors. Segmented scaling allows the continuity to be maintained, as long as scale factors are the same at the junctions. This is further supported by the availability of anthropometric data taken at segment junctions.

While the number of specifications used to describe human bodies is limited, they are still not necessarily satisfied by data from an anthropometric survey (Roebuck, 1995; Gordon et al.,

1989). More realistic scaling of human models requires better scaling algorithms as well as better survey data (Biferno, 1995; Roebuck, 1995).

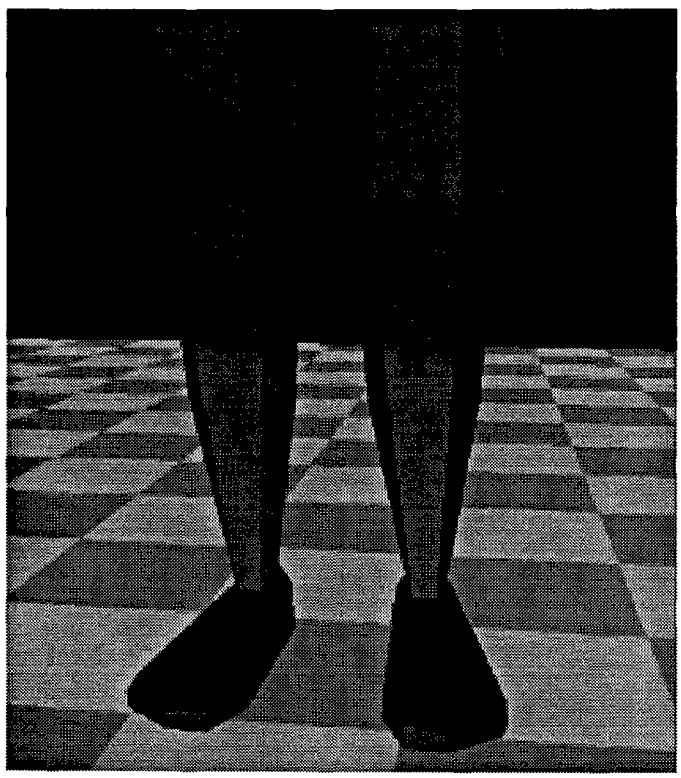

Figure 5. Anomalies caused by linear scaling.

Chapter 5

Anthropometry-Based Human Model Generation

FIGURE MODEL METHODS

A figure model may be implemented as an "object." An object is a collection of capabilities, plus the data associated with the implementation of those capabilities. From an anthropometric standpoint, a figure model has two major sets of capabilities, a set of figure-building methods and a set of anthropometric mensuration methods. Each of these sets is considered next.

Figure-Building Methods

Anthropometric figure building may be modeled as a mapping between data and figure model. There is not a one-to-one correspondence between anthropometric data and a geometric human figure model. In practice, the mapping is a composition of transformations such as regression computations, combination of measurements, geometric scaling operations, and so forth. We refer to all operations performed by this mapping as figure-building methods.

These methods provide the necessary capabilities to produce a figure instance sized according to a given set of (input) dimensions. To simplify the creation process (from a user point of view), a constructor method is provided. The constructor method can be seen as being a "master" method that takes anthropometric data as input and generates a figure object instance.

The figure-building methods are specific for a given figure model. They construct a figure, based on the type and number of geometric segments, the type and number of links, and the landmarks and joint centers defined in the model, the methods to perform the job of constructing a figure from these components.

However, it must be noted that these methods store no knowledge regarding the proportionality (preservation) aspects of the scaling. This information is assumed to be part of the input data. That is, all necessary statistical processing of the anthropometric data, including regression computations, estimations, and so forth, are assumed to take place as a step preceding the construction of the actual figure instance. With this pre-processed data, along with additional knowledge about anatomical aspects of the human body, as well as geometrical assumptions such as symmetry, the constructor produces a figure instance.

One of the major problems in traditional anthropometric models is that all the methods of the figure model are embedded inside a program. This approach suffers from serious problems. For example, if a component of the figure changes, it usually results in major overhauls of the program. Another deficiency of this approach is that the methods of the model are fixed and hidden from the user. If the user needed to adapt or fine tune the methods for a particular application, it would not be possible.

From an object-oriented point of view, the object's member functions are defined at the object's definition time. While the member functions of the object may remain fixed throughout the life of the object, an object-oriented framework allows for redefinition of such member functions (i.e., a member function taking the place of a default one).

From a figure model standpoint, one would benefit from an implementation of such an object-oriented framework. The benefits would be even greater, given an open architecture approach. Under this approach, a figure model is implemented as a "dumb" object, with no geometry, link structure, or methods. Nothing is attached to it except for a constructor method. The method provides a way to start the process of building the figure object instance. However, at instantiation time, the constructor expects all the necessary information and building methods to be supplied to it in order to construct the desired figure instance.

The Methods

In Jack, the figure model constructor takes the form of a stand-alone piece of software. The constructor module is available through the shell program "Genfig," which is discussed later in this document.

The definition of the figure model's topological structure is stored in a figure definition file. This file contains all details such as segments, articulations, attributes, and so forth. The figure definition file must be designed before any figure creation operation takes place. A default file is provided for the user's convenience. Specifications of an initial three-dimensional topological connection are provided. The figure file contains a so-called "generic" figure. From a geometric standpoint, it is complete in the sense that it can be displayed on a simulated environment created in the Jack graphical user interface. From an anthropometric point of view, it serves the purpose of defining the figure elements that are to be manipulated by the anthropometric building methods.

The task of the anthropometric building methods is to embed the anthropometric data into the generic figure. The constructor contains a set of primitive auxiliary methods whose purpose is to interpret rules. The collection of primitive auxiliary methods and interpreted rules defines the building methods. From a user standpoint, the primitive auxiliary methods are hidden in the figure constructor. The only thing the user need be concerned with is defining the rules.

The rules have been divided into two parts, according to an intermediate and target data representations abstraction. This abstraction helps isolate the purely anthropometric aspects of the data from the geometrical particularities of the figure model.

The intermediate data representation is basically a definition of anthropometric variables in terms of formulae based on ANSUR 88 measurements. The variable names are arbitrary, even though they are meant to remind us of the specific body segment they represent. Also, the axes associated with each definition are defined for breadth, depth, and length (x, y, z, respectively). Figure 6 shows a summary of the default version of this representation. The formulae are arithmetic expressions based on ANSUR 88 variables. The variable numbers[2] appear between square brackets. In other words, a number between square brackets is meant to be taken as a variable number, not literally. Also, some formulae of certain variables are noted in the Comment column as "guestimated," indicating that the variable's value is computed approximately, using a guessed factor[3]. Note that all these formulae are user redefinable.

The target representation of the rules uses the variables defined in the intermediate data representation to define variables associated with the (target) figure model. The latter variables are the ones ultimately used to specify the scaling of the figure model.

Figure 7 presents the target data representation. The following aspects must be noted. First, there is a recurring 0.5 scaling factor in two of the axes. The reason for the use of this factor is arbitrary and has to do with the way geometry is normalized (see Geometry Normalization in Appendix A).

---

[2]Notice that this numeration follows that of Cheverud et al. (1990). The reader is forewarned that in Gordon et al. (1989), the numeration has an offset of +1 with respect to the one presented here (i.e., each variable number appearing in the formulae must be incremented by 1).

[3]As with any aspect of the figure model definition, if better "guestimates" are available, it is easy to update the model.

Variable Name	Axis	Formula	Comment
clavicle	x	$(0.5 * [11])$	
upper_arm	x	$((1.0/\pi) * [12])$	$(\pi = 3.14159)$
upper_arm	y	$((1.0/\pi) * [12])$	
upper_arm	z	$([5])$	
lower_arm	x	$((1.0/\pi) * [53])$	
lower_arm	y	$((1.0/\pi) * [53])$	
lower_arm	z	$([88])$	
palm	x	$([58])$	
palm	y	$((2.0/3.0) * (1.0/\pi) * [127])$	((2.0/3.0) guestimate)
palm	z	$(0.6 * [60])$	((0.6) guestimate)
upper_leg	x	$([105])$	
upper_leg	y	$((1.0/\pi) * [104])$	
upper_leg	z	$(-[75] + [108])$	
lower_leg	x	$((1.0/\pi) * [29])$	
lower_leg	y	$((1.0/\pi) * [29])$	
lower_leg	z	$([75] - [76])$	
foot	x	$([76])$	
foot	y	$([51])$	
foot	z	$([10])$	

Variable Name	Axis	Formula	Comment
toes	x	$((1.0/3.0) * [76])$	(1.0/3.0 guestimate)
toes	y	$([51])$	
toes	z	$([52] - [10])$	
head	x	$([61])$	
head	y	$([63])$	
head	z	$([255])$	
neck	x	$((1.0/\pi) * [81])$	
neck	y	$((1.0/\pi) * [81])$	
neck	z	$(-[31] + [100] - [255])$	
upper_torso	x	$([33])$	
upper_torso	y	$([37])$	
upper_torso	z	$([31] - [103])$	
center_torso	x	$([113])$	
center_torso	y	$([116])$	
center_torso	z	$(-[68] + [103])$	
lower_torso	x	$([66])$	
lower_torso	y	$([25])$	
lower_torso	z	$([68] - [108])$	
finger	x	$(0.25 * [58])$	
finger	y	$(0.25 * [58])$	
finger	z	$(0.1333 * [60])$	(0.1333 guestimate)

Figure 6. Rules: Intermediate anthropometric data representation.

Segname.axis	Formula	Comment
t1.x	$0.5 * upper\_torso.x * 0.8166$	(0.8166 = 81.6% x.max)
t1.y	$(upper\_torso.z + center\_torso.z) * 0.04309$	(0.04309 = 4.3% of 1.0)
t1.z	$0.5 * upper\_torso.y * 0.7150$	(0.7150 = 71.5% z.max)
t2.x	$0.5 * upper\_torso.x * 0.9477$	idem
t2.y	$(upper\_torso.z + center\_torso.z) * 0.04567$	idem
t2.z	$0.5 * upper\_torso.y * 0.8600$	idem
t3.x	$0.5 * upper\_torso.x * 0.9812$	idem
t3.y	$(upper\_torso.z + center\_torso.z) * 0.04235$	idem
t3.z	$0.5 * upper\_torso.y * 0.9178$	idem
t4.x	$0.5 * upper\_torso.x * 1.0000$	(1.0000 = x.max)
t4.y	$(upper\_torso.z + center\_torso.z) * 0.06077$	idem
t4.z	$0.5 * upper\_torso.y * 0.8900$	idem
t5.x	$0.5 * upper\_torso.x * 0.9693$	(0.9603 = 96% x.max)
t5.y	$(upper\_torso.z + center\_torso.z) * 0.04899$	idem
t5.z	$0.5 * upper\_torso.y * 0.9612$	idem
t6.x	$0.5 * upper\_torso.x * 0.9693$	idem
t6.y	$(upper\_torso.z + center\_torso.z) * 0.05193$	idem
t6.z	$0.5 * upper\_torso.y * 0.9677$	idem
t7.x	$0.5 * upper\_torso.x * 0.9693$	idem
t7.y	$(upper\_torso.z + center\_torso.z) * 0.05193$	idem
t7.z	$0.5 * upper\_torso.y * 0.9852$	idem
t8.x	$0.5 * upper\_torso.x * 0.9693$	idem
t8.y	$(upper\_torso.z + center\_torso.z) * 0.05193$	idem
t8.z	$0.5 * upper\_torso.y * 0.9852$	idem
t9.x	$0.5 * upper\_torso.x * 0.9693$	idem
t9.y	$(upper\_torso.z + center\_torso.z) * 0.05488$	idem
t9.z	$0.5 * upper\_torso.y * 0.9900$	(0.9900 = z.max)
t10.x	$0.5 * upper\_torso.x * 0.9693$	idem
t10.y	$(upper\_torso.z + center\_torso.z) * 0.05967$	idem
t10.z	$0.5 * upper\_torso.y * 0.9825$	(0.9825 = 98.2% z.max)
t11.x	$0.5 * upper\_torso.x * 0.9442$	idem
t11.y	$(upper\_torso.z + center\_torso.z) * 0.03867$	idem
t11.z	$0.5 * upper\_torso.y * 0.9600$	idem
t12.x	$0.5 * upper\_torso.x * 0.9442$	idem
t12.y	$(upper\_torso.z + center\_torso.z) * 0.08361$	idem
t12.z	$0.5 * upper\_torso.y * 0.9350$	idem

Figure 7. Rules: Target anthropometric data representation.

Segname.axis	Formula	Comment
l1.x	$0.5 * upper\_torso.x * 0.9442$	(0.9442 = 95.4% x.max)
l1.y	$(upper\_torso.z + center\_torso.z) * 0.07845$	(0.07845 = 7.8% of 1.0)
l1.z	$0.5 * upper\_torso.y * 0.9000$	(0.9000 = 90% z.max)
l2.x	$0.5 * upper\_torso.x * 0.9275$	idem
l2.y	$(upper\_torso.z + center\_torso.z) * 0.07845$	idem
l2.z	$0.5 * upper\_torso.y * 0.9000$	idem
l3.x	$0.5 * upper\_torso.x * 0.9066$	idem
l3.y	$(upper\_torso.z + center\_torso.z) * 0.05782$	idem
l3.z	$0.5 * upper\_torso.y * 0.9000$	idem
l4.x	$0.5 * upper\_torso.x * 0.9066$	idem
l4.y	$(upper\_torso.z + center\_torso.z) * 0.07514$	idem
l4.z	$0.5 * upper\_torso.y * 0.9000$	idem
l5.x	$0.5 * upper\_torso.x * 0.9066$	idem
l5.y	$(upper\_torso.z + center\_torso.z) * 0.07661$	idem
l5.z	$0.5 * upper\_torso.y * 0.93$	idem

Segname.axis[.end]	Formula	Comment
bottom_head.x	$0.5 * (head.y)$	
bottom_head.y	$0.5 * (head.x)$	
bottom_head.z	$(head.z)$	
neck.x	$0.5 * (neck.y)$	
neck.y	$0.5 * (neck.x)$	
neck.z	$(neck.z)$	
upper_arm.1.x	$0.5 * (upper\_arm.y)$	
upper_arm.1.y	$0.5 * (upper\_arm.x)$	
upper_arm.1.z	$(upper\_arm.z)$	
upper_arm.2.x	$0.5 * (lower\_arm.y)$	
upper_arm.2.y	$0.5 * (lower\_arm.x)$	
upper_arm.2.z	1.0	arbitrary
lower_arm.1.x	$0.5 * (lower\_arm.y)$	
lower_arm.1.y	$0.5 * (lower\_arm.x)$	
lower_arm.1.z	$(lower\_arm.z)$	
lower_arm.2.x	$0.5 * (palm.x)$	
lower_arm.2.y	$(palm.z)/3.0$	(3.0 guestimate)
lower_arm.2.z	1.0	arbitrary
upper_leg.x	$0.5 * (upper\_leg.y)$	
upper_leg.y	$0.5 * (upper\_leg.x)$	
upper_leg.z	$(upper\_leg.z)$	
lower_leg.x	$0.5 * (lower\_leg.y)$	
lower_leg.y	$0.5 * (lower\_leg.x)$	
lower_leg.z	$(lower\_leg.z)$	
foot.x	$(foot.z)$	
foot.y	$0.5 * (foot.y)$	
foot.z	$0.5 * (foot.x)$	
toes.x	$(toes.z)$	
toes.y	$0.5 * (toes.y)$	
toes.z	$0.5 * (toes.x)$	
lower_torso.x	$0.5 * (lower\_torso.y)$	
lower_torso.y	$0.5 * (lower\_torso.x)$	
lower_torso.z	$(lower\_torso.z)$	
clavicle.x	$0.5 * (1.0) * 10.0$	1.0, 10.0 arbitrary
clavicle.y	$(1.0) * 10.0$	idem
clavicle.z	$0.5 * (clavicle.x) * 2.0$	2.0 cancels 0.5

Figure 7 (continued)

Segname.axis[.end]	Formula	Comment
palm.x	$0.5 * (palm.x)$	
palm.y	$(palm.z)$	
palm.z	$0.5 * (palm.y)$	
finger00.1.x	$0.5 * (finger.x)$	
finger00.1.y	$(finger.z)$	
finger00.1.z	$0.5 * (finger.y)$	
finger00.2.x	$0.5 * (finger.x)$	
finger00.2.y	1.0	arbitrary
finger00.2.z	$0.5 * (finger.y)$	
finger10.1.x	$0.5 * (finger.x)$	
finger10.1.y	$(finger.z)$	
finger10.1.z	$0.5 * (finger.y)$	
finger10.2.x	$0.5 * (finger.x)$	
finger10.2.y	1.0	arbitrary
finger10.2.z	$0.5 * (finger.y)$	
finger20.1.x	$0.5 * (finger.x)$	
finger20.1.y	$(finger.z)$	
finger20.1.z	$0.5 * (finger.y)$	
finger20.2.x	$0.5 * (finger.x)$	
finger20.2.y	1.0	arbitrary
finger20.2.z	$0.5 * (finger.y)$	
finger30.1.x	$0.5 * (finger.x)$	
finger30.1.y	$(finger.z)$	
finger30.1.z	$0.5 * (finger.y)$	
finger30.2.x	$0.5 * (finger.x)$	
finger30.2.y	1.0	arbitrary
finger30.2.z	$0.5 * (finger.y)$	

Segname.axis[.end]	Formula	Comment
finger01.1.x	$0.5 * (finger.x)$	
finger01.1.y	$(finger.z)$	
finger01.1.z	$0.5 * (finger.y)$	
finger01.2.x	$0.5 * (finger.x)$	
finger01.2.y	1.0	arbitrary
finger01.2.z	$0.5 * (finger.y)$	
finger11.1.x	$0.5 * (finger.x)$	
finger11.1.y	$(finger.z)$	
finger11.1.z	$0.5 * (finger.y)$	
finger11.2.x	$0.5 * (finger.x)$	
finger11.2.y	1.0	arbitrary
finger11.2.z	$0.5 * (finger.y)$	
finger21.1.x	$0.5 * (finger.x)$	
finger21.1.y	$(finger.z)$	
finger21.1.z	$0.5 * (finger.y)$	
finger21.2.x	$0.5 * (finger.x)$	
finger21.2.y	1.0	arbitrary
finger21.2.z	$0.5 * (finger.y)$	
finger31.1.x	$0.5 * (finger.x)$	
finger31.1.y	$(finger.z)$	
finger31.1.z	$0.5 * (finger.y)$	
finger31.2.x	$0.5 * (finger.x)$	
finger31.2.y	1.0	arbitrary
finger31.2.z	$0.5 * (finger.y)$	

Figure 7 (continued)

Segname.axis[.end]	Formula	Comment
finger02.x	$0.5 * (finger.x)$	
finger02.y	$(finger.z)$	
finger02.z	$0.5 * (finger.y)$	
finger12.x	$0.5 * (finger.x)$	
finger12.y	$(finger.z)$	
finger12.z	$0.5 * (finger.y)$	
finger22.x	$0.5 * (finger.x)$	
finger22.y	$(finger.z)$	
finger22.z	$0.5 * (finger.y)$	
finger32.x	$0.5 * (finger.x)$	
finger32.y	$(finger.z)$	
finger32.z	$0.5 * (finger.y)$	
thumb0.1.x	$0.5 * (finger.y)$	
thumb0.1.y	$(finger.z)$	
thumb0.1.z	$0.5 * (finger.x)$	
thumb0.2.x	$0.5 * (finger.y)$	
thumb0.2.y	1.0	arbitrary
thumb0.2.z	$0.5 * (finger.x)$	
thumb1.1.x	$0.5 * (finger.y)$	
thumb1.1.y	$(finger.z)$	
thumb1.1.z	$0.5 * (finger.x)$	
thumb1.2.x	$0.5 * (finger.y)$	
thumb1.2.y	1.0	arbitrary
thumb1.2.z	$0.5 * (finger.x)$	
thumb2.x	$0.5 * (finger.y)$	
thumb2.y	$(finger.z)$	
thumb2.z	$0.5 * (finger.x)$	

Figure 7 (continued)

In the segments associated with the torso, for the x and z directions, the largest value (segment) of the normalized geometry is 1.0. The other segments are given a factor equal to a percentage of that largest value. For the y direction (length), the y values of the normalized figure geometry sum to a total of 100%. Each y value of the normalized segments is divided by that 100% value, which results in the y factor.

Some segments have two declarations, differentiated by a ".1" or ".2" suffix. These cases denote the use of tapering scaling rather than bounding box scaling. As explained earlier in this document, tapering scaling considers both ends of the segment. Case .1 is associated with the proximal end and case .2 with the distal end. The formulas in Figure 7 contain variables in the form "variable.axis." These variables refer to the variables and axes of the intermediate representation.

Anthropometric Mensuration Methods

Aside from the figure-building capabilities, the figure model provides a set of anthropometric figure mensuration methods. These methods offer the capability of extracting the anthropometric information embedded in a figure model instance. It is important to realize that a figure model instance is, by itself, an entity representing the anthropometric data used to create it. If one places the figure instance in a graphics environment, its anthropometric dimensions (as defined by the figure model) are a representation of the actual anthropometric data within that simulated environment. However, it is necessary for the figure model to provide a way of extracting the measurements that have been embedded in it (and possibly some other extrapolated measurements). The mensuration methods provide this capability.

Primitive mensuration methods have been implemented in Jack. These primitive methods are complemented by (user) redefinable rules. A list of these rules appears next. These rules implement 69 of the measurements defined in the ANSUR 88 standard.

Mensurations are done on a site-to-site basis. The definition of sites, which are established on the (generic) figure definition file, can be as involved as necessary (to the extent of considering auxiliary planes, lines, etc.).

The necessary measurement entities (landmarks, sites, and planes) need to be defined in the figure model. In the default case, these entities were approximated visually when the generic figure was first defined. The resulting approximation is very rough. While no default rules are provided by the model to better approximate these entities, the system is prepared to accept such rules as input, should they be available. In other words, it is possible to use rules to define the location of any site in the figure model, including mensuration entities. (A plane, for example, may be defined using a site, since sites have orientation and position.)

As defined in the survey, most of the measurements are not vector distances but distances along a given axis. This is taken into consideration by specifying a measurement axis.

```
// FORMAT :
// P posturename     ---- defines a posture
//
// D variable_name variable# axis ---- distance definition
// sitel
// site2
//
// C variable_name variable# ---- circumference definition
// sitel
// site2
// site3
// site4
```

```
//
// ######## STANDING POSTURE ########
//
P mystand


D BIDLBDTH 13 0
right_upper_arm.LM_deltoid_pt_rt
left_upper_arm.LM_deltoid_pt_lft

D BIMBDTH 14 0
right_lower_leg.LM_lateral_malleolus
right_lower_leg.LM_medial_malleolus

D BIZBDTH 20 2
bottom_head.LM_right
bottom_head.LM_left

D BUTTDPTH 25 2
lower_torso.LM_buttock_pt_post
lower_torso.LM_front_pelvis

D CHSTBDTH 33 0
t7.LM_chest_right
t7.LM_chest_left

D CHSTDPTH 37 2
t7.LM_chest_back
t7.LM_chest_front

D HEADBRTH 61 0
bottom_head.LM_tragion_rt
bottom_head.LM_tragion_lft

D HIPBRTH 66 0
lower_torso.LM_buttock_pt_rt_lat
lower_torso.LM_buttock_pt_lft_lat

D BISBDTH 15 0
louer_torso.LM_hip_joint_rt
lower_torso.LM_hip_joint_lft

D INPUPBTH 69 0
right_eyeball.LM_pupil_rt
left_eyeball.LM_pupil_lft

D WSTBRTH 113 0
14.LM_waist_rt
14.LM_waist_lft

D WSTDPTH 116 2
14.LM_waist_post
14.LM_waist_ant_navel

D WRTHLGTH 131 2
right_lower_arm.LM_stylion
right_thumb2.LM_thumbtip
```

```
D INFORBB 240 2
bottom_head.LM_back
bottom_head.LM_infraorbitale_r

D ACRGHT 3 1
right_foot.LM_bottom_rfoot
right_upper_arm.LM_acromion_r

D CERVHT 31 1
right_foot.LM_bottom_rfoot
neck.LM_cervicale

D CRCHHCHT 39 1
right_foot.LM_bottom_rfoot
lower_torso.LM_crotch_level_rt

D ILCRSIT 68 1
right_foot.LM_bottom_rfoot
15.LM_iliocristale

D KNEEHTPMP 73 1
right_foot.LM_bottom_rfoot
right_lower_leg.LM_midpatella

D LATFEMEP 75 1
right_foot.LM_bottom_rfoot
right_upper_leg.LM_lat_femoral_epicondyle_standing

D LATMALHT 76 1 right_foot.LM_bottom_rfoot
right_lower_leg.LM_lateral_malleolus

D STATURE 100 1
right_foot.LM_bottom_rfoot
bottom_head.LM_TOP_HEAD

D SUPSTRHT 102 1
right_foot.LM_bottom_rfoot
tl.LM_suprasternale

D TENRIBHT 103 1
right_foot.LM_bottom_rfoot
13.LM_tenth_rib

D TROCHHT 108 1
right_foot.LM_bottom_rfoot
lower_torso.LM_trochanterion_rt

D WSTHNI 119 1
right_foot.LM_bottom_rfoot
14.LM_waist_rt

D ECTORBT 233 1
bottom_head.LM_ectoorbitale_rt
bottom_head.LM_TOP_HEAD

D TRAGT 255 1
bottom_head.LM_tragion_rt
bottom_head.LM_TOP_HEAD
```

```
D ACRDLG 5 3
right_upper_arm.LM_acromion_r
right_upper_arm.LM_radiale

D BLFTLG 10 3
right_foot.LM_pternion
right_foot.LM_first_metatarsophalangeal_protrusion

D BCRMBDTH 11 3
right_upper_arm.LM_acromion_r
left_upper_arm.LM_acromion_l

D FTBRHOR 51 3
right_foot.LM_first_metatarsophalangeal_protrusion
right_foot.LM_fifth_metatarsophalangeal_protrusion

D FOOTLG 52 3
right_foot.LM_pternion
right_toes.LM_acropodion

D HANDBRTH 58 3
right_finger00.LM_metacarpale_II
right_finger30.LM_metacarpale_V

D HANDLG 60 3
right_lower_arm.LM_stylion
right_finger12.LM_daclylion_III_r

D HEADLGTH 63 3
bottom_head.LM_glabella
bottom_head.LM_back

D INSCYE1 70 3
right_clavicle.LM_midscye_rt
left_clavicle.LM_midscye_lft

D RASTL 88 3
right_upper_arm.LM_radiale
right_lower_arm.LM_stylion

D SHOULGTH 93 3
right_upper_arm.LM_acromion_r
neck.LM_trapezius_pt_rt

D WRINFINGL 130 3
right_lower_arm.LM_stylion
right_finger02.LM_dactylion_II

// --------- Circumferences ---------
C ANKLCIRC 6
right_lower_leg.LM_ankle_back
right_lower_leg.LM_ankle_front
right_louer_leg.LM_ankle_left
right_lower_leg.LM_ankle_right

C CALFCIRC 29
right_louer_leg.LM_back
right_louer_leg.LM_front
```

```
right_lower_leg.LM_left
right_louer_leg.LM_right

C KNEECIRC 72
right_louer_leg.LM_midpatella
right_louer_leg.LM_midp_back
right_louer_leg.LM_midp_left
right_lower_leg.LM_midp_right

C NECKCIRC 81
neck.LM_back
neck.LM_front
neck.LM_neck_rt_lat
neck.LM_neck_lft_lat

C THGHCIRC 104
right_upper_leg.LM_back
right_upper_leg.LM_front
right_upper_leg.LM_left
right_upper_leg.LM_right

C WRISCIRC 127
right_lower_arm.LM_styl_back
right_lower_arm.LM_styl_front
right_lower_arm.LM_styl_left
right_lower_arm.LM_styl_right

//
// ######## SITTING POSTURE ########
//
P mysit

D BUTTKLTH 27 2
lower_torso.LM_buttock_pt_post
right_lower_leg.LM_knee_pt_ant

D BUTTPLTH 28 2
louer_torso.LM_buttock_pt_post
right_upper_leg.LM_popliteal_sit

D ACRHTST 4 1
lower_torso.LM_sit_level
right_upper_arm.LM_acromion_r

D CERVSIT 32 1
lower_torso.LM_sit_level
neck.LM_cervicale

D EYEHTSIT 50 1
lower_torso.LM_sit_level
bottom_head.LM_ectocanthus

D KNEEHTSIT 74 1
right_foot.LM_bottom_rfoot
right_lower_leg.LM_suprapatella

D POPHGHT 87 1
right_foot.LM_bottom_rfoot
```

```
right_upper_leg.LM_popliteal_sit

D SITTHGHT 94 1
lower_torso.LM_sit_level
bottom_head.LM_TOP_HEAD

D THGHCLR 105 1
lower_torso.LM_sit_level
right_upper_leg.LM_thigh_front

D NSTHSTNI 121 1
lower_torso.LM_sit_level
14.LM_waist_rt

D WSHTSTOM 122 1
lower_torso.LM_sit_level
14.LM_waist_ant_navel

//
// ######## FUNCTIONAL POSTURE ########
//
P functionalzero

D THMBTPR 107 2
t4.LM_back_pt_rt
right_thumb2.LM_thumbtip

D HRHALLLN 132 2
t4.LM_back_pt_rt
right_lower_arm.LM_stylion

P functionalone

D WRHALLEX 133 2
t4.LM_back_pt_rt
right_lower_arm.LM_stylion

//
// ######## SITTING FOREARM-UP ########
//
P sitting_forearm_up
D lower_torso.LM_sit_level right_lower_arm.LM_olecranon_flex 1 49 ELRHGHT

//
// ######## STAND FOREARM-UP  ########
//
P forearm_up

D FORHDLG 55 2
right_lower_arm.LM_olecranon_flex
right_finger12.LM_daclylion_III_r

D SHOUELLT 92 3
right_upper_arm.LM_acromion_r
right_lower_arm.LM_olecranon_flex

//
// ######## FOREARM-UP-HAND-CLOSED ######
```

```
//
P forearm_up_hand_closed

D WRCTRGRL 126 2
right_lower_arm.LM_stylion
right_palm.LM_hand_closed_center

//
// ######## BICEPS FLEX #######
//
P biceps-flex
// --------- Circumferences ---------

C BICIRCFL 12
right_upper_arm.LM_back
right_upper_arm.LM_front
right_upper_arm.LM_left
right_upper_arm.LM_right

C FCIRCFL 53
right_lower_arm.LM_back
right_lower_arm.LM_front
right_lower_arm.LM_left
right_lower_arm.LM_right

// // ######### OVERHEAD POSTURE ########
//
P overhead
D OVHDFTRH 84 1
right_foot.LM_bottom_rfoot
right_finger12.LM_dactylion_III_r

//
// ######### ARM SPAN POSTURE ########
//
P arm_span
D SPAN 99 0
right_finger12.LM_dactylion_III_r
left_finger12.LM_dacrylion_III_l
```

Error in Measurements

Circumferences are estimated from the following equation:

$$circumf(d_1,d_2) = \pi \cdot \sqrt{\left(\frac{d_1^2 + d_2^2}{2}\right)}$$

in which d_1,d_2 are the major and minor axes (or vice versa) in the measurement's cross section. Note that this is an approximation for "circumference" anthropometric measurements.

As for linear measurements, the following aspects introduce error:

- Location of measurement sites.

  While the measurement sites have been carefully defined, it must be understood that the geometry of the body does not exhibit a one-to-one correspondence to a real human being. The shape of the torso, for instance, is as simple as possible to reduce the polygonal complexity of the figure. Definition of measurement sites must take this into account.

- The type of geometry used to model the body.

  Rigid (i.e., non-deformable) geometries are employed. Therefore, sites like "lower_torso.LM_sit_level" can only be defined approximately to account for any deformations.

- Measurements that span several segments can show inaccuracies. Assuming that a normalization technique is employed for scaling, where the distance between joint centers is normalized, the only linear measurements that are guaranteed are link lengths, as defined in Appendix B. Lengths other than link lengths, involving more than one link, cannot be guaranteed. There are various reasons for this:

  - The mechanical joint model used in the body does not necessarily correspond to a real human joint. For instance, the shoulder complex is modeled with two mechanical joints. This offers limited similarity to an actual human shoulder complex. Accordingly, the accuracy of a functional reach, which is a measurement of maximum reach span and usually involves the shoulder complex, is limited by the accuracy of the motion of the simulated shoulder complex. (See Appendix C for pictures of all postures used for anthropometric measurements.)

  - Measurements spanning several links (e.g., buttocks-popliteal length) usually consider soft tissue (body fat, skin). Soft tissue is not modeled by the polybody, except of the geometrical characteristics of the surfaces themselves. The scaling process used for the sizing of the polyhedral surfaces does not consider overall shape but only link length and certain circumferences. Beyond that, the enfleshment of the polybody is provided by the generic (artistically modeled) surfaces.

  - Somatotype modeling is not considered. To understand what this implies, consider scaling the body using an average person's measurements. If the shape of the geometry used to model the body is fixed (at design time), then problems may be encountered when trying to model an obese (or a thin) person with the same geometry used to model an average size person.

JOINTS

A joint is defined using a site from each of two body segments to be connected. Figures 8 and 9 show sites and joints currently defined in Jack.

Each joint in the human body has a range of motion. The range of motion of a joint described in terms of angles is measured in degrees for each degree of freedom (DOF). That is to say, for each plane in which movement is allowed at a joint, there is a range of motion.

The human figure model of Jack allows motion at 68 joints which have a total of 135 DOF. For each DOF, two measurements are required, upper and lower limits. That means that there are 270 joint measurements for each human figure. The joints' default data are extracted from NASA (1987), Louis (1983), and Chaffin and Andersson (1991).

The following list shows the corresponding degrees of freedom for each joint. The associated data (not shown) are extracted from NASA (1987) and Chaffin and Andersson (1991). Spine joint limits are extracted from Louis (1983). Other values not appearing in this list have been "guestimated."

```
NECK (U. Limit)
x     neck lateral right
y     neck flexion
z     neck rotation right
NECK (L. Limit)
x     neck lateral left
y     neck extension
z     neck rotation left
SHOULDER (U. Limit)
x     shoulder, abduction
y     shoulder, flexion
z     shoulder, rotation lat
SHOULDER (L. Limit)
x     shoulder, adJuction (from O.B. by D. Chaffin)
y     shoulder, extension
z     shoulder, rotation med
ELBOW (U. Limit)
y     elbow, flexion
ELBOW (L. Limit)
y     elbow, extension
WRIST (U. Limit)
x     wrist radial
y     wrist flexion
z     forearm pronation
WRIST (L. Limit)
x     wrist ulnar
y     wrist extension
z     forearm supination
HIP (U. Bimit)
x     hip abduction
y     hip flexion
z     hip lateral rotation prone (O.B. by D. Chaffin2
HIP (L. Limit)
x     hip adduction (O.B. by D. Chaffin)
y     hip extension
```

```
z       hip medial rotation prone (O.B. by D. Chaffin)
KNEE (U. Limit)
y       knee flexion
KNEE (L. Limit)
y       knee extension
ANKLE (U. Limit)
y       ankle, dorsi
ANKLE (L. Limit)
y       ankle, planter
KNUCKLES (U. Limit)
y       flexion
KNUCKLES (L. Limit)
y       extension
```

J nt Name	*Site-A*	*Site-B*	*Type*
left_eyeball	bottom_head.left_eyeball	left_eyeball.base	xz
right_eyeball	bottom_head.right_eyeball	right_eyeball.base	xz
atlanto_occipital	neck.distal	bottom_head.proximal	zyx
base_of_neck	t1.distal	neck.proximal	yzx
solar_plexus	t1.distal	upper_torso.proximal	NONE
left_clavicle_joint	upper_torso.lclav	left_clavicle.proximal	xy
right_clavicle_joint	upper_torso.rclav	right_clavicle.proximal	-xy
left_shoulder	left_clavicle.lateral	left_upper_arm.proximal	zxy
right_shoulder	right_clavicle.lateral	right_upper_arm.proximal	-z-xy
left_elbow	left_upper_arm.distal	left_lower_arm.proximal	y
right_elbow	right_upper_arm.distal	right_lower_arm.proximal	y
spinet2t1	t2.distal	t1.proximal	xyz
spinet3t2	t3.distal	t2.proximal	xyz
spinet4t3	t4.distal	t3.proximal	xyz
spinet5t4	t5.distal	t4.proximal	xyz
spinet6t5	t6.distal	t5.proximal	xyz
spinet7t6	t7.distal	t6.proximal	xyz
spinet8t7	t8.distal	t7.proximal	xyz
spinet9t8	t9.distal	t8.proximal	xyz
spinet10t9	t10.distal	t9.proximal	xyz
spinet11t10	t11.distal	t10.proximal	xyz
spinet12t11	t12.distal	t11.proximal	xyz
spinel1t12	l1.distal	t12.proximal	xyz
spinel2l1	l2.distal	l1.proximal	xyz
spinel3l2	l3.distal	l2.proximal	xyz
spinel4l3	l4.distal	l3.proximal	xyz
spinel5l4	l5.distal	l4.proximal	xyz
waist	lower_torso.distal	l5.proximal	yzx
left_hip	lower_torso.lhip_lateral	left_upper_leg.proximal	zxy
right_hip	lower_torso.rhip_lateral	right_upper_leg.proximal	-z-xy
left_knee	left_upper_leg.distal	left_lower_leg.proximal	-y
right_knee	right_upper_leg.distal	right_lower_leg.proximal	-y
left_ankle	left_lower_leg.distal	left_foot.proximal	zxy
right_ankle	right_lower_leg.distal	right_foot.proximal	-z-xy
left_toes	left_foot.toes	left_toes.proximal	y
right_toes	right_foot.toes	right_toes.proximal	y

Figure 8. Joint description (hands in Figure 9).

Joint Name	*Site-A*	*Site-B*	*Type*
left_wrist	left_lower_arm.distal	left_palm.base	yxz
right_wrist	right_lower_arm.distal	right_palm.base	y-x-z
linfinger02	left_finger01.tip	left_finger02.base0	-x
linfinger01	left_finger00.tip	left_finger01.base0	-x
lmidfinger12	left_finger11.tip	left_finger12.base0	- x
lmidfinger11	left_finger10.tip	left_finger11.base0	- x
lringfinger22	left_finger21.tip	left_finger22.base0	-x
lringfinger21	left_finger20.tip	left_finger21.base0	-x
lpinfinger31	left_finger30.tip	left_finger31.base0	- x
lpinfinger32	left_finger31.tip	left_finger32.base0	- x
left_finger00	left_palm.f11	left_finger00.base0	z-x
left_finger10	left_palm.f22	left_finger10.base0	z-x
left_finger20	left_palm.f33	left_finger20.base0	z-x
left_finger30	left_palm.f44	left_finger30.base0	z-x
lthumb2	left_thumb1.tip	left_thumb2.base0	-x
lthumb1	left_thumb0.tip	left_thumb1.base0	-x
lthumb0	left_palm.thumb0	left_thumb0.base0	-zy
rinfinger02	right_finger01.tip	right_finger02.base0	x
rinfinger01	right_finger00.tip	right_finger01.base0	x
rmidfinger12	right_finger11.tip	right_finger12.base0	x
rmidfinger11	right_finger10.tip	right_finger11.base0	x
rringfinger22	right_finger21.tip	right_finger22.base0	x
rringfinger21	right_finger20.tip	right_finger21.base0	x
rpinfinger31	right_finger30.tip	right_finger31.base0	x
rpinfinger32	right_finger31.tip	right_finger32.base0	x
right_finger00	right_palm.f11	right_finger00.base0	zx
right_finger10	right_palm.f22	right_finger10.base0	zx
right_finger20	right_palm.f33	right_finger20.base0	zx
right_finger30	right_palm.f44	right_finger30.base0	zx
rthumb2	right_thumb1.tip	right_thumb2.base0	x
rthumb1	right_thumb0.tip	right_thumb1.base0	x
rthumb0	right_palm.thumb0	right_thumb0.base0	-z-y

Figure 9. Joint description (hands).

MASS

The global mass is the sum of the masses of all body segments. In the Jack human model, the mass of each segment is computed, according to Webb Associates (1978), as a percentage of the total mass (gblmass). The percentages are average percentile values for a fit male population as found in the NASA male crew member trainees. For the average general population or a population skewed to small or lightweight to large or heavyweight, these percentages will vary. The formula

used to compute each segment's mass appears to the right of the segment's name. These formulae are redefinable.

```
bottom_head             0.0790*gblmass
left_ eyeball           0.0001*gblmass
right_eyeball           0.0001*gblmass
neck                    0.0020*gblmass
left_clavicle           0.0300*gblmass
right_ clavicle         0.0300*gblmass
t1                      0.3128*gblmass/17
t2                      0.3128*gblmass/17
t3                      0.3128*gblmass/17
t4                      0.3128*gblmass/17
t5                      0.3128*gblmass/17
t6                      0.3128*gblmass/17
t7                      0.3128*gblmass/17
tS                      0.3128*gblmass/17
t9                      O.3128*gblmass/17
t10                     0.3128*gblmass/17
t11                     0.3128*gblmass/17
t12                     0.3128*gblmass/17
11                      0.3128*gblmass/17
12                      0.3128*gblmass/17
13                      0.3128*gblmass/17
14                      0.3128*gblmass/17
15                      0.3128*gblmass/17
left_upper_arm          0.0280*gblmass
right_upper_arm         0.0280*gblmass
left_lower_arm          0.0160*gblmass
right_lower_arm         0.0160*gblmass
left_palm               0.0040*gblmass
right_palm              0.0040*gblmass
left_finger32           0.0020*gblmass/15
left_finger31           0.0020*gblmass/15
left_finger30           0.0020*gblmass/15
left_finger22           0.0020*gblmass/15
left_finger21           0.0020*gblmass/15
left_finger20           0.0020*gblmass/15
left_finger12           0.0020*gblmass/15
left_finger11           0.0020*gblmass/15
left_finger10           0.0020*gblmass/15
left_finger02           0.0020*gblmass/15
left_finger01           0.0020*gblmass/15
left_finger00           0.0020*gblmass/15
left_thumb2             0.0020*gblmass/15
left_thumb1             0.0020*gblmass/15
left_thumb0             0.0020*gblmass/15
right_thumb2            0.0020*gblmass/15
right_thumb1            0.0020*gblmass/15
right_thumb0            0.0020*gblmass/15
right_finger32          0.0020*gblmass/15
right_finger31          0.0020*gblmass/15
right_finger30          0.0020*gblmass/15
right_finger22          0.0020*gblmass/15
right_finger21          0.0020*gblmass/15
right_finger20          0.0020*gblmass/15
right_finger12          0.0020*gblmass/15
```

```
right_finger11          0.0020*gblmass/15
right_finger10          0.0020*gblmass/15
right_finger02          0.0020*gblmass/15
right_finger01          0.0020*gblmass/15
right_finger00          0.0020*gblmass/15
lower_torso             0.1260*gblmass
left_upper_leg          0.1000*gblmass
right_upper_leg         0.1000*gblmass
left_lower_leg          0.0460*gblmass
left_lower_leg          0.0460*ghlmass
right_lower_ leg        0.0460*gblmass
left_foot               0.0126*gblmass
right_foot              0.0126*gblmass
left_toes               0.0014*gblmass
right_toes              0.0014*gblmass
```

CENTER OF MASS COMPUTATION

The center of mass of the segments in the figure is computed based on Webb Associates (1978) and NASA (1987) (as shown in Figure 10) and the segment's surface geometry.

Segment	X-axis	Y-axis	Z-axis
Head	0.5 * Bitragion Breadth	Tragion-Wall Depth	0.17 * Tragion Vertex Ht
Up. Arm	assume Symmetry	assume Symmetry	0.48 * Link Length
Lo. Arm	assume Symmetry	assume Symmetry	0.41 * Link Length
Hand	Symmetry at z-axis loc.	Symmetry at z-axis loc.	0.51 * palm length
Up. Leg	assume Symmetry	assume Symmetry	0.41 * Link Length
Lo. Leg	assume Symmetry	assume Symmetry	0.44 * Link Length
Foot	Symmetry at z-axis loc.	Symmetry at z-axis loc.	0.44 * foot length (from heel)

Figure 10. Segment center of mass computation.

SPECIAL BODY CONSTRUCTS

There are four "special" body constructs:

- Shoulder Complex. For a detailed explanation, refer to Badler, Phillips, & Webber (1993) and Zhao (1993).

- Torso-Spine. This model is considered in detail in Monheit & Badler (1991) and Badler, Phillips, & Webber (1993).

- Hand. This model is presented in the next section of this document.

- Foot "Complex". The foot is modeled with two (rigid) segments, (a) base of foot or ball of foot (defined from heel to ball of foot), and (b) toes (defined from ball of foot to

toe tips). The toes segment shows no toe detail (neither segments nor joints). For the foot complex, there are two joints, ankle and toes. The former connects the base of the foot with the lower leg segment. That latter connects the base of the foot to the toes segment. See Appendix D for sites used for normalization of segments.

ANTHROPOMETRY OF THE HAND

The human model includes a detailed model of the hands. This modeling has been done in accordance with Anderson (1983), Brand (1985), and McMinn & Hutchings (1977). The fingers are modeled with three segments each and three joints. The phalanges include proximal, middle, and distal segments. The modeling of the thumb includes the first metacarpal segment, so that this finger has three segments like the other digits. The joints of the fingers are as follow:

- trapezio-metacarpal joint between the metacarpal segment of the thumb and the palm, with two degrees of freedom (extension-flexion and adduction-abduction),

- metacarpophalangeal joint between the palm and the phalangeal proximal segment of the digit, with two degrees of freedom (extension-flexion and adduction-abduction). This joint in the thumb has only one degree of freedom (extension-flexion).

- interphalangeal joint between the proximal and middle, and the middle and distal phalanges. There is one degree of freedom (extension-flexion).

The palm geometry has been shortened in order to portray (from a skeletal point of view) the proper connection sites with the digits. This results in a hand that appears to have "long" fingers and a "short" palm. This hand is more similar to the skeletal counterpart and biomechanically more correct.

Even though this approach gives generally good results, several points must be made:

- In a real human hand, the proximal phalange of digit 2 (the index finger's proximal phalange) starts at a location closer to the wrist than the proximal phalanges of digits 3, 4, and 5.

- The palm model used in the human figure assumes that digits 2, 3, 4, and 5 start at the same location (i.e., are collinear).

- The overall result of making this assumption is that digit 2 will, sometimes, appear larger than digit 3. This is generally incorrect.

The palm consists of one segment with six joints, one for each digit and the wrist. Carpals or metacarpals are not modeled. (While the carpal segment of the thumb is modeled, it is considered part of the thumb and not of the palm.)

Scaling

Scaling of the hand is based on Greiner (1991). Figure 11 describes the mapping between the anthropometric data and the geometry of the hand. Circumference dimensions may be obtained by dividing by π.

Hand Joint Limits

Joint limits of the fingers are based on Berme, Paul, and Purves (1977); Brand (1985); Cooney, Lucca, Chao, and Linsheid (1981); and Joseph (1951). The limits can been seen in Figure 12.

These references consider range of motion, rather than joint limits. The data are reported as a total range. This range has been divided into upper and lower limits, based on educated guesses and empirical observations. Future data will require more accuracy.

FEMALE

Currently, there is no geometry for the female human body model (i.e., female polybody). To overcome this, the male body can be scaled according to female anthropometric values. In addition, the appearance of the head can be feminized by substituting female for male features.

Obviously, this is not the best solution, as it led to errors because of sexual differences in proportionality and segment alignment (e.g., hip breadth, breasts, etc.). A "native" female polybody will be available in the future.

Note that the default mass factors are for a male population (see Mass section).

Segment/Digit	x/y	z
Finger 0 (digit2)	-	10
0	13,14	21
1	13,14	20
2	15,16	19
Finger 1 (digit3)	-	22
0	25,26	33
1	25,26	32
2	27,28	31
Finger 2 (digit4)	-	34
0	37,38	45
1	37,38	44
2	39,40	43
Finger 3 (digit5)	-	46
0	49,50	57
1	49,50	56
2	51,52	55
Thumb (digit1)	-	01
0	4,5	07
1	4,5	08
2	4,5	09

Figure 11. Anthropometric survey data mapping.

Joint	Range of Motion (degrees)
Finger DIP	Ext-Flx(60)
Finger PIP	Ext-Flx(100)
Finger MCP	Ext-Flx(90) Abd-Add(60)
Thumb IP	Ext-Flx(85)
Thumb MCP	Ext-Flx(50)
Thumb CMC	Ext-Flx(50) Abd-Add(40)

Figure 12. Joint limits.

Chapter 6

Statistical Methods

INTRODUCTION

This chapter presents statistical methods for population data manipulation. For manipulation of anthropometric data, three methods available are Monte Carlo family simulation, cadre family simulation, and multinormal conditional estimation of proportionality patterns.

PROPORTIONALITY RECONSTRUCTION

The notion of proportionality can be easily understood if we think of the (body) proportions of a particular individual. However, the concept of proportionality preservation for a population cannot be easily compared with that for an individual. The proportionality for each subject will vary across a population. Yet, a given proportionality will always be identified with a single subject. However, for a population, we will use "statistical" subjects (also known as synthetic subjects). A statistical subject is an artificial entity that represents a given aspect of the population. Furthermore, rather than considering individual statistical subjects, we are interested in families of such subjects, as explained in the next section.

From a statistics standpoint, proportionality preservation indicates the observance of correlations between variables of interest, when applying a given statistical model. For our purposes, let us consider proportionality preservation in two ways. First, we are interested in preservation of proportions, when using population sample data to scale a figure model. Second, we want to be able to reconstruct data necessary for scaling a figure model, starting from incomplete sets of anthropometric data. We will eventually discuss preservation of proportions when we discuss figure families. At this moment, we turn to discussing proportionality reconstruction.

Reconstruction Method

In the following discussion, we assume the reader has a basic knowledge of the multinormal distribution model. (For further details, refer to Morrison, 1967.)

Assuming that the covariance matrix of the multinormal distribution is full rank (nonsingular), the following properties hold:

• Given an N-dimensional random vector $\bar{X}$, with a multinormal distribution, mean $\bar{\mu}$, and covariance matrix Σ of rank N. For any $M \cdot N$ real matrix Q, with rank $M \leq N$, the vector $\bar{A} = Q\bar{X}$ is a multinormal random variable. Furthermore, its mean is $E(\bar{A}) = Q\bar{\mu}$ and its covariance matrix is $Q\Sigma Q^T$.

• Given a $(p + q)$ dimensional multinormal population, with variates represented by the random vector $\bar{X} = [\bar{X}_0, \bar{X}_1]$, in which $\bar{X}_0$ is of size p and $\bar{X}_1$ is of size q. The mean vector is

$$\bar{\mu} = [\bar{\mu}_0, \bar{\mu}_1]^T$$

and the covariance matrix is

$$\Sigma = \begin{bmatrix} \Sigma_{00} \Sigma_{01} \\ \Sigma_{10} \Sigma_{11} \end{bmatrix}$$

in which Σ_{00} has dimensions $p \cdot p$, Σ_{01} has dimensions $p \cdot q$, and Σ_{11} has dimensions $q \cdot q$. Also, $\Sigma_{10} = \Sigma_{01}$. $\bar{X}_0$ and $\bar{X}_1$ are multinormal random vectors. The distributions of these vectors are $N(\bar{\mu}_0, \Sigma_{00})$ and $N(\bar{\mu}_1, \Sigma_{11})$, respectively. Σ_{01}'s values, relative to those of Σ_{00} and Σ_{11}, determine the degree and pattern dependence between the two sets of variates.

• A random vector $\bar{X}$ (partitioned into two random sub-vectors $\bar{X}_0$ and $\bar{X}_1$), with multivariate normal distributions $\bar{\mu}$ and Σ, will be independently distributed if and only if $\Sigma_{01} = 0$.

• The conditional density function of X_0, assuming elements of $\bar{X}_1$ are fixed to $\bar{x}_1$, is expressed as

$$g(\bar{x}_0 \mid \bar{x}_1) = \frac{f(\bar{x}_0, \bar{x}_1)}{h(\bar{x}_1)}$$

in which f represents the joint density function of the complete set of $p + q$ variates and h is the joint density function of the q fixed variates.

• It can be proved (Morrison, 1967) that the conditional distribution of a set of p variates from a multinormal population, with q other variates of the population held fixed, is multinormal with (conditional) mean vector

$$\bar{\mu}_c = \bar{\mu}_0 + \Sigma_{01}\Sigma_{11}^{-1}(\bar{x}_1 - \bar{\mu}_1)$$

And (conditional) covariance matrix

$$\Sigma_c = \Sigma_{00} - \Sigma_{11}^{-1}\Sigma_{01}^{T}$$

The implications of the above results are as follow:

- Given a p-dimensional multinormal random population with mean $\bar{\mu}$ and covariance matrix Σ, and given a random vector $\bar{X}_1$ of size q, extracted from that population, so that $q \leq p$, it is possible to estimate a random vector $\bar{X}_0$, with dimension $p - q$, based on the given (fixed) q values, thus obtaining a (complete) p-dimensional model, with $\bar{X}_0$ having mean $\bar{\mu}_c$ and covariance matrix Σ_c and $\bar{X}_1$ having mean $\bar{\mu}$ and covariance matrix Σ.

- Therefore, if we have *a priori* knowledge of the correlations of a p-dimensional multinormal random population and extract q (key) scaling features out of an individual from that population, we can estimate the remaining $p - q$ values.

FAMILIES OF FIGURES

In work space design problems, it is common to encounter multivariate situations in which the correlations among variables need to be taken into consideration. The traditional (but unfortunately incorrect) practice of considering (univariate) percentile-based data in the construction (scaling) of figure models leads to undesirable results, such as the trio of all 5th, 50th, and 95th percentile dummies.

The purpose of a population accommodation test is to verify that all individuals in a given population fit a given accommodation problem. An accommodation percentage is the fraction of all individuals tested that actually fit the environment. (If the Monte Carlo simulation is used, the purpose of such a test is to verify that all elements of a sample with appropriate size fit a given accommodation problem. The size is determined so that the simulation converges to a solution.) The problem with this approach is that the number of individuals that must be tested is equal to the population (sample) size. (For Monte Carlo simulations, that size is significantly large.)

Another approach, referred to as boundary cases, considers only "extreme" points along the hypersurface of the multinormal distribution of the given population. Such an approach is significant, not only because it reduces the number of individuals needed for testing, but because it focuses attention on "important" cases. In practice, Monte Carlo and cadre families complement each other.

In the next section, we review both the Monte Carlo and the population "extremes" approach. However, before we proceed in that direction, let us briefly explain the notion of a "statistical individual."

A statistical individual is an abstraction that consolidates a number of anthropometric variables across a given population. A population (statistical) individual is produced as a result of a statistical transformation of the population data. When one performs statistical transformations of anthropometric data, the individuality of each observation (population subject) disappears, to lead toward general or common traits that define the population. It is necessary to use proportionality preserving statistical transformations so that correlations among variables are considered.

Monte Carlo Simulation

The multivariate correlation model can be applied in simulations of body proportionality and space requirements. These simulations fall under the type known as "Monte Carlo" (Ulam & Metropolis, 1949).

Controlled randomness is the key to these simulations. The generation of human figure model anthropometric parameters is randomly generated, hoping that one particular instance fits a set of desired design constraints.

Algorithmic randomness simulates real world randomness. This randomness is controlled using correlations obtained from real world population data, thus making the stochastic process approximate the "randomness" of the real world.

When a random vector of correlated variables is generated to simulate, for example, the dimensional parameters of the human body, a set of (probabilistic) requirements is being satisfied. Yet, one can claim that only with a certain probability in N experiments does a generated pattern satisfy the constraints being considered. It could be a first try, the last one, all or none. Monte Carlo simulations have been applied to the problem of space accommodation (Bittner & Moroney, 1974; Bittner, 1976). In this context, N experiments are conducted and tested against a given set of spatial requirements. Statistics are obtained from the results of this simulation to ascertain what percentage of the population is accommodated within the space undergoing analysis.

Given access to raw population data, one could try the following alternate analysis. Randomly pick samples of N individuals from the population and test their measurements against the given space constraints. Then, obtain statistics of the results to determine the size of the accommodated population. One can determine appropriate sampling sizes (N) that can provide, with a given probability, satisfactory results.

Roebuck (1995, p. 96) states:

> The result of [a Monte Carlo simulation] is...a large set of synthetic operators (typically 400-3600), each having dimensions of varying percentiles within the manikin design and each dimension statistically distributed with essentially the same means and standard deviations and with all pairs of dimensions correlated to the same degree of the original population. So, although no single synthetic operator is guaranteed to match a living subject, each operator represents one possible case that could occur in a population of living people, without violating any of the underlying statistics.

Implementing a Monte Carlo Family

Let us assume that the covariance matrix Σ is symmetric positive definite (Bronson, 1989). Since every square matrix Q is similar to a matrix J in Jordan canonical form, and if M is a modal matrix for Q, then for the square covariance matrix Σ, it holds that

$$\Sigma = MJM^{-1}$$

in which J is a diagonal matrix of the eigenvalues of Σ and M is composed of the eigenvectors of Σ. Therefore, $MM^{-1} = \mathrm{I}$. In fact, being the covariance matrix symmetric positive definite, $M^{-1} = M^T$.

Let us define a random variable

$$\overline{A} = M^T \overline{X}$$

in which $\overline{A}$ is a multinormal random variable, with mean $E(\overline{A}) = M^T \overline{\mu}$ and covariance matrix $M^T \Sigma M$. Since $\Sigma = MJM^T$, then it follows that the covariance matrix of $\overline{A}$ is J.

We are fabricating a (whitening) transformation that yields a random vector with covariance equal to the identity matrix (i.e., a random vector whose variates are not only independent but have a unitary standard deviation).

Observing that the square root of a positive definite matrix Q is given by

$$Q^{1/2} = MJ^{1/2}M^{-1}$$

and noticing that, if Q is diagonal, then

$$Q^{1/2} = J^{1/2}.$$

Finally, let us define

$$\overline{B} = J^{-1/2}\,\overline{A}$$

in which $\overline{B}$ is a multinormal random variable with mean $E(\overline{B}) = J^{1/2}M^T\,\overline{\mu}$ and covariance matrix $J^{-1/2}M^T\Sigma M(J^{-1/2})^T = J^{-1/2}M^TMJ^{-1}M^TM(J^{-1/2})^T = \mathrm{I}$. Now, suppose that we generate a standardized (multinormal) random vector $\overline{\delta}$ (Box & Muller, 1958; Muller, 1959), with mean $\overline{\mu} = 0$ and covariance matrix $\Sigma = \mathrm{I}$. Then, by means of the inverse of the transformation just defined, we can transform that random vector into another one with a specific mean and covariance. It can be proved easily that given a random vector $\overline{\delta}$, with mean $\overline{\mu} = 0$ and covariance matrix $\Sigma = \mathrm{I}$, and applying the transformation

$$\overline{X} = MJ^{1/2}\,\overline{\delta}$$

we obtain a multinormal random vector $\overline{X}$ with covariance matrix $\Sigma = MJM^T$. Furthermore, if we add the mean vector $\overline{\mu}$ to $\overline{X}$, we obtain a multinormal random vector with mean $\overline{\mu}$. Each such vector represents a member of the Monte Carlo family.

The Cadre Family

The cadre[4] family (Bittner, Glenn, Harris, Iavecchia, & Wherry, 1987), also known as the boundary family, is an alternate approach to the Monte Carlo approach. Bittner et al. (1987) demonstrates that, in general, for accommodation testing purposes, the cadre family is at least as good as its Monte Carlo counterpart. The cadre family exhibits features such as having the ability to screen the multivariate space in a systematic fashion and the ability to substantially reduce the size of testing space, while capturing a significant amount of variance. This goal is accomplished by systematically considering only extreme cases of the population.

Regarding boundary manikins, Roebuck (1995, p. 95) states, "These boundary conditions are much better able (sic) to guarantee a given percentage of accommodation in design than are common-percentile manikins."

[4]From the French for frame, limits, scope.

It is important to understand several aspects regarding the cadre family:

• A cadre family is a multivariate representation of the extremes of the population distribution. For example, given *n* variables, the cadre family consists of $2^n + 2n + 1$ subjects. The problem is that, as *n* becomes big (>7), we observe a combinatorial explosion of extreme cases. To "delay" the occurrence of such an explosion of combinations, instead of using the variables directly in the application of the method, we employ the principal factors with larger associated variance, capturing a significant percentage of the total variation.

• The cadre family is a "statistical" representation of the extremes of a population. This implies that none of the cadre members may actually have a real-world counterpart. This is an issue that should be carefully analyzed when building mock-ups of the cadre family simulation.

• Each of the family members represents an orthogonal trait. A trait can be understood as a linear combination of (some of) the original variables. This characteristic may be useful when using the family for accommodation design purposes. However, making sense of the traits in terms of the original variables is generally not easy to do.

Implementing a Cadre Family

Our implementation considers Bittner et al. (1987) and Meindl, Zehner, & Hudson (1993). For details about principal components and the multivariate normal distribution, refer to Morrison (1967).

The probability density function of the multivariate normal distribution is defined by

$$f(\bar{x}) = [(2\pi)^p |\Sigma|)^{-1/2} exp(-1/2(\bar{x} - \bar{\mu})^T \Sigma^{-1}(\bar{x} - \bar{\mu})]$$

in which *p* is the number of dimensions, $\bar{x}$ is a random vector, and $\bar{\mu}$ and Σ are the mean and variance-covariance matrix of the population, respectively.

Any symmetric matrix (e.g., Σ) has an associated quadratic form, $Q(\bar{x})$. The quadratic form is defined by

$$Q(\bar{x}) = \bar{x}^T \Sigma \bar{x}$$

By equating this quadratic form with a constant value, *k*, the resulting expression defines a conic surface. The shape of the conic surface depends on the diagonal elements of Σ. For Σ representing a variance-covariance matrix, the conic surface described is a hyperellipsoid.

Vector $\bar{x}$ defines specific locations of the surface of the hyperellipsoid. For the case of $p = 2$, the case of an ellipse, vector $\bar{x}$ is collinear with the gradient vector of $Q(\bar{x})$ at four directions. These directions are those of the principal axes of the ellipse. In general, for any given p, vector $\bar{x}$ is collinear with the gradient vector of $Q(\bar{x})$ at $2 \cdot p$ directions. The key points about the principal components are

- The computation of the principal components is done so that each of the components has a correlation of 0 with the other components.

- The variances associated with the principal components are not all equal to 1. (Note that the original variables have variance equal to 1.) The sum of the variance of the principal components will be p, the number of variables. However, the components are computed so that the first component captures the largest variance, the second one captures the next largest variance, and so on. As a consequence, a few components capture most of the variance. This fact can be used to justify the canceling of components that do not contribute significantly to the total variance.

The generation of the cadre family involves the following steps:

- The value of constant k, defining the hyperellipsoid size, must be determined for a given accommodation percentage. This basically translates to finding the value of the constant k of the quadratic form defined previously. Meindl et al. (1993) suggest finding this value by iteration.

- The next step in the solution is to find the component scores. This is done by computing the principal components of the covariance matrix.

The principal components, P_i ($i = 1,...,k$), are linear combinations of the original variables. P_1 has the largest sample variance, P_2 has the next largest variance, and so on:

$$P_i = a_{1,1}Z_1 + a_{2,1}Z_2 + ... + a_{p,1}Z_p$$

for p variables $Z_1, Z_2, \ldots, Z_p$. Computation of the weight coefficients $a_{i,j}$ is constrained by

$$\sum_{k=1}^{p} a_{k,j}^2 = 1$$

The components are prioritized according to their associated captured variance (descending order), and as many are kept as needed for the sum of the associated variances to equal a desired total captured variance.

- In the next step, the component scores matrix is multiplied by a binary matrix formed by all combinations of $\pm$ k. Also, one may want to compute the axial points. This can be done by forming a matrix defining, in the same row, a single +k and a -k in two adjacent columns, with each row having exactly two elements different than 0. This step yields standard score vectors.

- The final step is to convert the standard scores into raw scores.

The family generated should include, not only the hyperellipsoid surface points, but also the axial points and the mean. The total number of subjects produced is $2^m + 2m + 1$, in which m is the number of factor loadings as determined by the minimum amount of variance specified.

Our implementation supports input in the form of raw data; or correlations, means, and standard deviations; or covariances and means. By establishing a series of parameters such as accommodation percentage and percentage of variance sought, a given number of cadre subjects is automatically determined.

Chapter 7

Generator of Figures

GENFIG

The Genfig (generator of figures) program marks the start of a new way of thinking about human figure modeling. The program departs from the style of its predecessor, spreadsheet system for anthropometric scaling (SASS), toward a more user-friendly, anthropometric and multivariate statistics-based framework.

Following the spirit of SASS, Genfig offers a combination of capabilities that allow the user to construct anthropometrically based human figure models. Genfig introduces new ideas in anthropometric modeling. Genfig is not an updated version of SASS. Genfig has been designed and implemented as an entirely new program. In fact, Genfig is only a shell program that presents a front end to a set of stand-alone modules. In this way, new modules can be easily added to Genfig using a "plug and play" approach. For example, the figure constructor module, currently designed for handling the construction of polybody figure instances, could easily be replaced by another module designed for handling the construction of smooth skin figure instances.

Currently, there are four plug and play modules in Genfig:

- Figure constructor handles the creation of peabody, grammar-based figure definition files, following the rules presented in Chapter 5.

- Solver implements the statistical conditional reconstruction process presented in the previous chapter.

- Monte Carlo family generator implements the production of Monte Carlo families of figures, as explained in the previous chapter.

- Cadre family generator implements the production of a cadre family, as explained in the preceding chapter.

Each of these modules is a stand-alone program that can be run from the command line. Genfig has no knowledge about the internal implementation of these modules. All it knows is what the interface (parameter list) of each program exposes. In this way, it is easy to see how the implementation of any of these modules can be modified, as long as the interface remains intact.

In the future, the user will be allowed to add new modules into the functionality of Genfig or maybe acquire them from a third party vendor. Currently, however, it is only possible to replace any of the existing modules by comparable ones (same interface). For example, a new constructor module handling smooth skin figures could replace the default one.

Genfig is implemented as a stand-alone module complementing the Jack software package. Figure 13 shows the X-windows based user interface of the program.

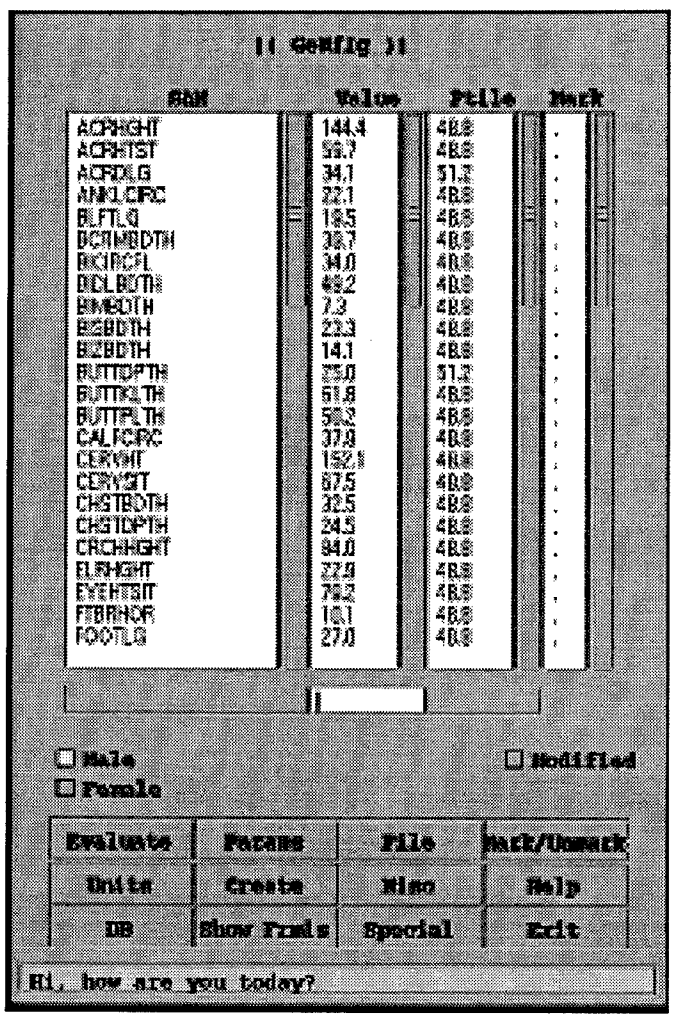

Figure 13. Graphical user interface of Genfig.

Additional information about Genfig is presented in its user manual (to be published as part of the Jack's User Manual). Also, refer to the previous chapter of this document for detailed information about the statistical methods. The next two sections present a brief overview of the two main methods associated with figure modeling, that is, the constructor and the measurer. The former is one of the modules in Genfig. The latter is implemented as a module in Jack.

THE CONSTRUCTOR

The rules used by the constructor method have been explained in Chapter 5 and are not repeated here. The constructor module takes anthropometric data (see the list of default variables in Appendix E). All the variable declaration data associated with those variables and methods to handle those data, which are required by the constructor to build figure model instances, are expected as input to the program. The module provides defaults for all these inputs stored in plain text files. The module has been designed to work in an interpreted fashion. Each time a figure instance is created, the program interprets the methods (or rules) and data needed for the job. Interpretation is not inefficient, as might be expected. In fact, the constructor has been optimized to be about 30 times faster than the figure generator in SASS. Furthermore, writing scripts for the program is not cumbersome. A traditional interpreted language, such as LISP, was originally considered for scripting, but developers used a much simpler (yet powerful) script format based exclusively on variable declarations and arithmetic expressions.

The output of the constructor module is a figure definition file. (See Appendix F for an example.)

THE MEASURER

In addition to the Genfig program, the anthropometric measurement capabilities in Jack have been changed to support the functionality of Genfig. The measurement methods of the figure model are implemented in Jack.

The measurements are done all at once, through the execution of the Jack command measure all. As a result of this command, a file with the name of the figure being measured, but with the extension ".dmp." This file contains the value for each of the variables defined in the measurement rules. The list of the default measurement rules has been presented in Chapter 5. As with all other aspects of Genfig, the set of default measurement rules is fully user redefinable. In this way, the rules can be updated or easily replaced.

Chapter 8

Anthropometric Error

ANTHROPOMETRIC ERROR MODEL

In this chapter, a brief overview of anthropometric error is presented. For a more in-depth discussion, refer to Azuola (1996).

Anthropometric Error

To address the problem of quantifying the accuracy of the approximations used in the application of anthropometric, one-dimensional data to the construction of a three-dimensional figure model, the notion of anthropometric error, *AE*, is introduced. We define anthropometric error as

$$AE = D_2 - D_1$$

in which D_2 is the value of an anthropometric measurement extracted from a human figure instance, and D_1 is the anthropometric measurement value used for scaling that figure instance.

Causes of AE

Anthropometric error in a human figure can have three main causes:

- Scaling data. For example, the one-dimensional nature of anthropometric measurements imposes limitations on the usability of such data in the construction of anthropometric figures. Another example is the use of univariate percentile data, rather than multivariate data, in the construction of the figure instances.

- Geometry and link structure. As explained earlier in this document, anthropometric human models are limited by the degree of control offered by the geometry (shape and sizing). Limited control leads to approximations, which themselves lead to error.

- Approximation equations used to match data and geometry/link structure. Consider the approximation equations used in building the polybody figure in Chapter 5. These equations handle only a small subset of all anthropometric measurements available in the ANSUR-88 survey. As their name indicates, evaluating these equations results in approximate values, each of which corresponds to a specific anthropometric parameter used in the construction of the figure instances. The use of approximations leads to error.

Note that in the case of Genfig, it is very easy to modify, enhance, or even replace the default set of equations with a set of better equations. Further research should be done to develop higher quality approximation equations.

AE Model

By itself, the *AE* associated with the different anthropometric variables is a passive entity, namely, a collection of error values. An *AE* model is an abstraction that considers *AE* not only from the standpoint of a collection of values but also introduces data measurement and analysis methods. The implementation of the *AE* model results in a tool designed for testing the anthropometric error of both specific individual figures or of entire families of figures.

As a result of such tests, an $m \cdot n$ anthropometric error matrix, E, is produced, in which m is the number of variables tested and n is the size of the testing family (an individual can be considered a degenerate case of a family of one member). The error matrix, E, presents the error for each of the variables and each of the figures involved in the testing.

The *AE* measurement model tool provides some basic data analysis methods. However, the user is expected to have access to a spreadsheet tool or a statistical analysis package for more detailed analysis of the data, if deemed necessary. In any case, the measurement tool handles the following functions:

- Conversion of E into $|E|$. This function allows the user to convert all entries of E into positive values. A negative entry in E means that the measured value on the figure model is smaller than the control value. This information is important, but on certain occasions, what is needed is not knowing the direction of the error, but its magnitude. Ideally, the matrix E should be 0. Matrix $|E|$ allows us to better see how far E is from the ideal without concern for signs. For example, if we compute the mean of a given row of E, the negative and positive values may cancel each other and result in a mean close to 0, which would give us the incorrect impression that the population *AE* for that particular row is close to ideal. Using $|E|$, on the other hand, results in a better understanding of the actual size of the mean *AE* for a given row.

- Data filtering. This function detects which rows (variables) are affected by the different figures in the population. It yields E_R, a reduced version of E, that contains only the rows (variables) which are affected by the different figures. For example, if the value of variable X is constant across all figures in the family being tested, then the row for X is not included in E_R. This allows the user to concentrate only on the variables that are affected by the figure type.

- Means, standard deviations, and correlations. These are basic statistics and correlations that summarize the data in *E*. The correlations can tell us how the *AE*'s are correlated with each other. After all, the *AE*'s, themselves, follow a multivariate distribution.

The *AE* modeling tool uses the Monte Carlo and cadre families presented earlier in this document. The use of these families in the framework of *AE* differs somewhat from their use in work space design. In the *AE* framework, the use of these families of figures permits *AE*-based testing of the scalability of a given figure model, to all the possible combinations found in a given anthropometric family. The (data) members of these families are scaling patterns (vectors), one per member. Each member is tested by using it in the scaling of the given figure model, and then, the *AE*'s of the resulting figure instance are measured. While the members of the Monte Carlo family represent random cases of the corresponding multinormal distribution, the cadre family members represent the extreme cases of such distribution. Using the cadre family for *AE*-based testing of the scalability of the figure model allows us to determine the anthropometric quality of the members of the resulting figure family, each of which represents a worst case scaling condition.

Chapter 9

Inverse Dynamics

INTRODUCTION

The inverse dynamics code in Jack has been rewritten, in an effort to expand its capabilities and add flexibility. The new interface allows the user to pick an arbitrary part of any articulated figure (including, but not limited to, the standard human figure), and compute forces at the joints for both static and dynamic postures. Force information can now be obtained for any part of the articulated human spine, arms, and legs. Simulated external forces (such as extra weight) can be applied anywhere on the figure to analyze their effect on the joint forces.

INTRODUCTION TO INVERSE DYNAMICS

Inverse dynamics is the calculation of the forces required at a human figure's joints in order to produce a given set of joint accelerations. The term "force" is used to refer to both translational forces and rotational forces (torques). Static figure postures are a special case where the joint accelerations are all zero. However, even in the static case, forces at the joints exist to counteract the external forces acting on the figure (such as gravitational force).

For the inverse dynamics calculations, all the external forces acting on a figure should be known. One does not need to worry about the weight of the segments, since it is automatically computed by Jack. Other forces, most commonly contact forces between figure segments and the environment, are generally unknown or difficult to compute. The solution is to define a root site for the part of the figure of interest. Think of the root as the point where the figure is attached to the world. The rest of the figure segments are hinged in a tree structure from the root through the joints. If there are no other contacts between the part of the figure we are interested in and the environment, then nothing else need be done. For example, if one is studying the forces at the joints in the upper human body, as long as no other upper body segment is attached to the environment, the root can be set at the pelvis. There is no problem if there are external forces, all of which are known, as the root can be placed in any segment.

The placement of the root defines an internal tree hierarchy in the figure. Each segment is connected to its parent via a joint. Inverse dynamics computes the torques at the joints as well as the force acting on a segment from the segment's parent. The magnitude of this force is

particularly important in reference to evaluating the stress on the lower human back, given a particular posture.

The most efficient way to calculate inverse dynamics is to use the recursive Newton-Euler dynamic equations. Jack's implementation of them is based on the work of Roy Featherstone (1987).

INVERSE DYNAMICS IN JACK

Inverse dynamics in Jack allow the user to compute static and dynamic joint forces for stationary and moving postures, respectively. The user can select the part of the body that is of interest, assign the root to any site on the figure, and apply external forces on any figure segment. Motions can be created using the standard Jack motion generation commands. The joint velocities and accelerations, as well as the translational velocity and acceleration of the root site, are used to compute the dynamic joint forces.

Torques at the joint are computed around the defined joint axes. Forces acting on a segment are displayed both in the local segment frame (segment coordinates) as well as in the inertial frame (world coordinates). The units used throughout the calculations are kilograms, meters, and seconds.

All the available commands can be executed from the force submenu of Jack or can be typed directly just like any other job control language command.

COMMANDS

This section describes the commands used to access the inverse dynamics force data. They are given in the force submenu of Jack.

create force data - This command is used to activate the inverse dynamics system. Knowing which joints are of interest in a particular joint force calculation, the user defines the tree structure that contains all of them by selecting the root site (described previously) and any number of leaf segments. Only segments contained in the paths from the root to the leaves will be part of force calculation. As an example, to measure joint forces for the right hand of a human figure, one might select `right_clavicle_lateral` as a starting site and `right.palm` as an ending segment. Alternatively, if one is interested in all upper body forces, one might choose `lower_torso_proximal` for the start site and `right.palm` and `left.palm` for the end segments. To execute the command, the user has to press "Esc" after the last segment is entered. All the segments that become part of the force calculations are displayed in the Log window. Multiple

parts of the figure can be marked for inverse dynamics calculations by repeatedly executing this command.

delete force data - This command resets all previously defined inverse dynamics tree structures for a figure.

display static joint torque - This command prints the static joint torques at a particular joint. Static joint torques refer to the torques needed to maintain a posture when the figure has zero velocity and acceleration. (The only forces acting are the weight and any external force the user defined on the figure.)

display static segment force - This command displays the three-dimensional force that a segment is receiving from its parent in the tree hierarchy. For example, to obtain the force on the lowest segment of the human spine, one might ask for the force on the fifth lumbar segment, 15. The force is displayed for convenience both in the local coordinate system of the segment and in the inertial (world) coordinate system.

display dynamic joint torque and display dynamic segment force - In order to analyze the joint forces when a figure is in motion, one has to follow these steps:

- Use create force data to define the part of the figure that is of interest.
- Create any motions on the figure using the standard Jack motion commands.
- Type Go to execute the motion.
- Use the animation window slider or goto time to select a particular figure posture.
- Run display dynamic joint torque or display dynamic segment force to get the joint forces at the particular time instant.

The velocities and accelerations of the root site and the joints are used to calculate the joint forces. Velocities and accelerations are computed under the assumption that there are 30 Jack frames per actual time second. One can adjust the frame rate through the set frame rate command.

attach force to site - This command sets a simulated external force on a segment. The force is applied through a particular segment site, and its direction should be given in the world coordinate system. For example, to attach an additional weight *w* to a site, one should enter *0, w, 0.* (Note that the weight of an object is equal to its mass in kilograms multiplied by the gravitational acceleration 9.81 m/s^2.)

clear segment forces - This command clears all external forces from a segment (except from the segment's weight, of course).

READING FRAME DATA

Determining the dynamic joint forces relies on getting the figure and joint velocities and acceleration from standard Jack channels. The most straightforward way to fill channels with data is to use standard Jack motion commands. If, however, the motion is inserted from precomputed frames through the read frames command, then information about the root site channel should also be read. To obtain the correct structure of a frame file, one should use create force data, then generate a simple motion and write the channel data into a file using write frames. The root site channel data will be written in the frame file as well.

Chapter 10

Smooth Skin

The current human model used in the center is based on rigid segments. Such a rigid body model provides the ability for real-time manipulation because of its computational simplicity. However, a rigid body model is inadequate to provide a visually satisfying image, mainly because the connection between segments often creates discontinuities. To correct this unwanted property, we have designed a control system to deform the geometry.

The subsequent development of computer graphics has provided different tools in modeling deformable material. In this work, "free-form deformation" (Sederberg & Parry, 1986; Parry, 1986) was chosen to control the human geometry deformation.

DESIGN GOAL

A common misconception with a deformable human body is that the deformation has to be volume conservative. However, this conservation property does not hold true because of the non-homogeneity of human tissue and the complexity of human body structure. For example, the change of shape and size for muscle flexed by a body builder will differ from that for a regular person. Another example would be the change in body volume between inhalation and exhalation. This kind of muscle behavior and volume are case dependent and beyond the scope of the work reported herein. Instead, our design goals are focused on smooth geometry transition between segments around the joint area and real time performance.

Smooth Geometry Transition of a Joint

To model a deformable body, one has to decide how the deformation will be assigned. Since every joint in a human body connects two segments, the deformation is distributed between these two segments. As mentioned before, our work does not concern material properties. It is therefore reasonable to assume that each segment contributes half of the deformation. That is, if the rotation angle of the joint is θ, the joint cross section rotation angle is $\theta/2$.

Real-Time Performance

One way to control the geometry deformation is to treat the whole body as a contiguous segment. Accordingly, the change of human posture is totally controlled by manipulating the control mesh. Although this method guarantees smooth transition between joints, it is very laborious. It applies the deformation algorithm on every node whenever a joint is moved.

To solve this problem, we perform geometry segmentation to minimize the application of the deformation algorithm. With geometry segmentation, when a joint is moved, the deformation algorithm only applies to those segments for which deformation actually occurs. Other segments simply perform rigid body motion.

GEOMETRY SEGMENTATION

Segmentation of the current rigid human body model is based on skeletal structure. With this type of segmentation, the model can best simulate skeletal motion. However, for the purpose of simulating geometry deformation, geometry segmentation is based more on muscular than on skeletal structure.

In human anatomy, a skeleton is divided into axial and appendicular sections (Tortora & Anagnostakos, 1993). As the muscular structure differs between these two sections, so does deformation.

The axial skeleton consists of pelvis, torso, neck, and head. In the rigid body model, each vertebra is considered as one unique segment. However, the muscles usually cross over several vertebra segments. When the muscle groups contract (or relax), all these segments are deformed. The result of deformation should be a continuous surface that describes the shape of the muscle groups. To ensure smoothness between segments, we use one contiguous segment to simulate the whole axial section.

The appendicular skeleton contains the bones of free appendages, which consist of upper extremities (arms), lower extremities (legs), and girdles that connect to the axial section. In contrast to axial muscles, appendicular muscles only cross one joint. The deformation caused by these muscles is usually limited to the area around the joint. Therefore, the geometry segmentation follows the articulatory segmentation. As might be expected, appendicular deformation is much easier to simulate than axial deformation.

ALGORITHM FOR FREE-FORM DEFORMATION

The first decision to implement free-form deformation is to choose the embedded algorithm and the knot vector. Commonly used algorithms include Bezier, piecewise Bezier, B-spline, and the nonuniform rational B-spline (NURB[5])-based system (Farin, 1990; Piegl, 1990). The Bezier algorithm is the simplest algorithm to implement. However, it connects every control node with every segment node. When the control mesh size is increased, computation increases as well,

[5]a three-dimensional projection of a four-dimensional curve

becoming too expansive. In contrast, the NURB system offers the most flexible control of free-form deformation. However, some of the NURB control options are redundant in this implementation. For our system, we use closed end, cubic, B-spline-based, free-form deformation. The knots in this implementation are equally spaced.

CONTROL

Control is the most important issue in implementing a deformable model. For our implementation, we have adopted the layer construction concept (Chadwick, Haumann, & Parent, 1989) to build the control mechanism. At the higher level, the user changes the human posture by adjusting the joints between virtual skeleton segments. When the displacements of these joints are changed, the behavior level control mechanism moves the free-form deformation control nodes based on the control algorithm and the new displacements. After that, the segment geometry is decided according to the new locations of the control nodes and the embedded free-form deformation algorithm.

The behavior control level is hidden from the user. It is more complex than the higher level and controls the actual deformation of the geometry, based on the joint angles. As the muscular structures differ between axial and appendicular sections, let us consider them separately.

Axial Division

There are two types of axial joints, dependent and independent. Joints connecting spine segments are dependent. Because of the muscular structure, these joints always move as a group. Joints connecting clavicle segments are independent, as they can move independently from the spine joints. However, since the muscle groups that control the clavicle segments' movements are connected to all 12 vertebrae, the geometry of the clavicle segments are also influenced by spine joints.

In our implementation, we have assigned two control meshes to the axial section. The lower mesh controls the lower torso in which the geometry changes only when the spine joints change. The upper control mesh controls the upper part of the torso. In this region, both spine movement and shoulder movement will affect geometry deformation.

Appendicular Division

The appendicular skeletal structure contains various types of joints, including ball, saddle,

hinge, and pivot (Tortora & Anagnostakos, 1993). They have either one degree or three degrees of freedom. Different types of joints create different types of skeletal movement. However, with the movement masked by muscle groups wrapped around the skeleton system, geometry deformations for different types of joints do not significantly differ.

There are two types of deformation effects, bending and twisting. Bending occurs when the rotation axis is perpendicular to the principal axis of the geometry. Twisting occurs when the rotation axis is parallel to the principal axis of the geometry. These two types deform different parts of the geometry. In our implementation, we have assigned three deformation meshes onto each appendicular segment. Two end meshes, located over the joint area, are used to simulate bending. Another mesh, placed in the center of the geometry, is used to control twisting.

RESULT

Figure 14 shows the result of our implementation. The left-hand figure demonstrates deformation of the appendicular section. Examples of bending of the axial division appear in the center and on the right.

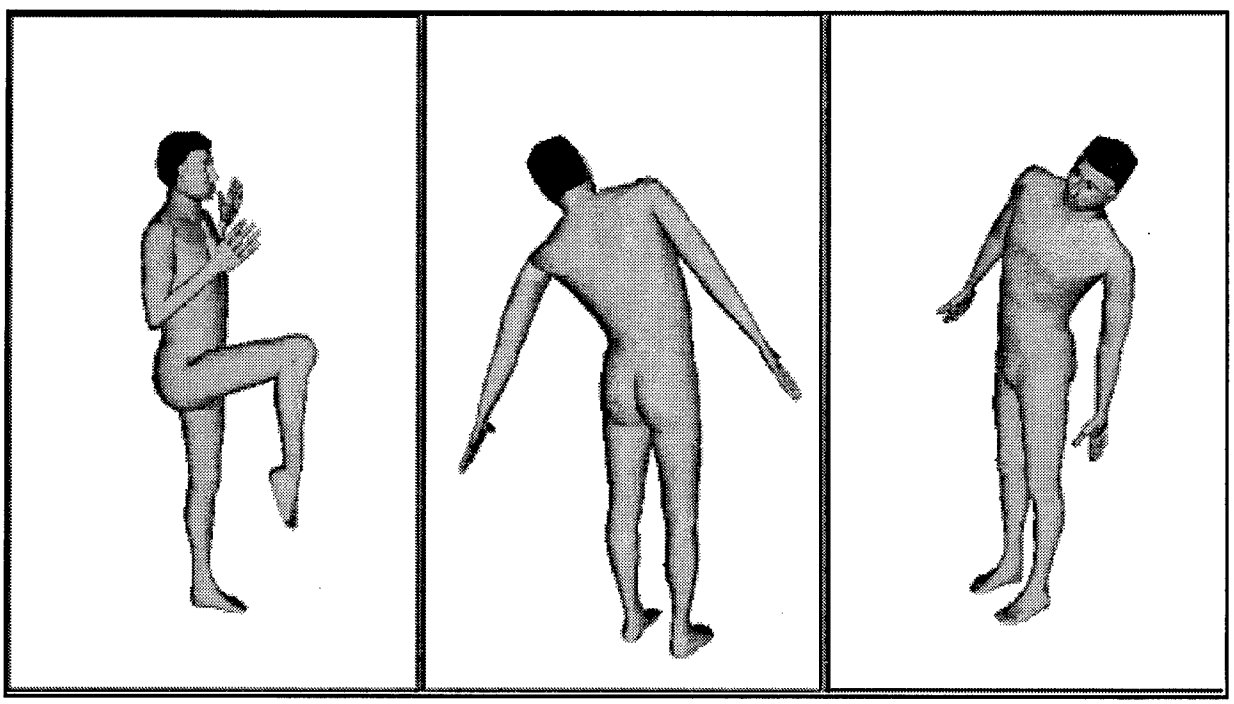

Figure 14. Deformable human model.

Chapter 11

Estimation of Landmark Locations

As was stated in Chapter 1, there are no three-dimensional landmark locations as determined by a traditional anthropometric survey. Educated guesses are necessary to estimate these locations. (Appendix G is a list of landmarks for the human model.) The following are some criteria used to place landmarks on the current model. All landmarks on the model are user redefinable.

1. lower_torso

a. LM_abdominal_pt: There is no abdomen of the figure, so the center point on the most protruding face of the lower stomach of the sitting figure is assumed to be the abdominal point.

b. LM_buttock_pt: This is a point along the middle face of the leg at the level of the buttock point.

c. LM_gluteal_furrow_pt: Since there is no distinct thigh in the figure, assume that the gluteal furrow point is the center point where the lower face of the buttock comes into the leg.

2. upper_arm

a. LM_acromion: There is no direct line from the neck. Use the skeleton to match to the highest point on the lateral tip of the shoulder.

b. LM_biceps_pt: Since the figure does not flex, assume that the biceps point is 20% more in the x direction than unflexed.

3. toes

a. LM_acropodion: Since there are no toes on the figure, the point farthest away from the heel is assumed to be the acropodion.

4. clavicle

a. LM_axillary_fold_post: There is no axillary posterior fold, so it is estimated to be directly under the clavicle.

b. LM_clavicle_pt: The point on the figure that corresponds with the clavicle point on the skeleton.

5. neck

a. LM_cervicale: the corresponding edge on the neck of the figure with the seventh cervical vertebra.

6. Bottom_head

a. LM_chin: most protruding point in the center of the chin.

b. LM_ear_bottom: the lowest protruding point on the earlobe.

c. LM_ear_top: the highest protruding point on the ear.

d. LM_ectocanthus: There are no eyelids on the figure, so the outer corner where the top and bottom of the eye come together is assumed to be the ectocanthus.

e. LM ectoorbitale: the point on the figure that corresponds with the ectoorbitale on the skeleton.

f. LM_gonion: point on the figure that corresponds with the point on the skeleton.

7. finger

a. LM_dactylion_II: the outermost tip of the right index finger.

b. LM_dactylion_III: the outermost tip of the middle finger.

8. upper_leg

a. LM_dorsal_juncture: the center point where the two planes from the calf and the thigh intersect.

9. lower_arm

a. LM_elbow_crease: when the elbow is flexed, the center point where the two faces of the forearm and the upper arm intersect.

10. foot

a. LM_fifth_metatarsophalangel_protrusion: the point on the figure that corresponds with the skeleton.

b. LM_first_metatarsophalangel_protrusion: the point on the figure that corresponds with the skeleton.

REFERENCES

Alameldin, T. (1991). Three dimensional workspace visualization for redundant articulated chains. Unpublished doctoral dissertation, University of Pennsylvania.

Anderson, J.E. (1983). Grant's atlas of anatomy (8th ed.). Baltimore: Williams and Wilkins.

Azuola, F. (1995). SASS Version 2.5 user's manual. Philadelphia: University of Pennsylvania, Department of Computer and Information Science.

Azuola, F. (1996). Anthropometric error in computer-based human figure models. Unpublished doctoral dissertation, University of Pennsylvania, Philadelphia.

Badler, N.I., Phillips, C.B., & Webber, B.L. (1993). Simulating humans, computer graphics animation and control. New York: Oxford University Press.

Bailey, D.A., Malina, R.M., & Rasmussen, R.L. (1978). The influence of exercise, physical activity, and athletic performance on the dynamics of human growth. In F. Falkner & J.M. Tanner (Eds.), Human growth, Volume 2. New York: Plenum Press.

Basmajian, J.V. (1982). Primary anatomy (8th ed.). Baltimore: Williams and Wilkins.

Berme, N., Paul, J.P., & Purves, W.K. (1977). A biomechanical analysis of the metacarpophalangeal joint. Journal of Biomechanics, 10, 409-412.

Biferno, M. (October 1995). Human modeling technology and standards. In M. Biferno (Chair), G-13 Committee, Society of Automotive Engineers Meeting, San Diego.

Bittner, A.C. (1976, June). Computerized accommodated percentage evaluation: review and prospectus. Proceedings of 6th Congress of the International Ergonomics Association, 11-16.

Bittner, A.C., Glenn, F.A. III, Harris, R., Iavecchia, H.P., & Wherry, R.J. (1987). Cadre: a family of manikins for workstation design. In Trends in ergonomics/human factors IV (pp. 733-740). Amsterdam: North Holland.

Bittner, A.C., & Moroney, W.F. (1974). The accommodated proportion of a potential user population: Compilation and comparisons of methods for estimation. Proceedings of Human Factors Society 1974, 376-381.

Box, G.E.P., & Muller, M.E. (1958). A note on the generation of random normal deviates. Annals of Mathematical Statistics, 29,. 610-611.

Brand, P.W. (1985). Clinical mechanics of the hand. St. Louis: Mosby.

Bronson, R. (1989). Schaum's outline of theory and problems of matrix operations. New York: McGraw-Hill.

Chadwick, J.E., Haumann, D.R., & Parent, R.E. (1989). Layered construction for deformable animated characters. Computer Graphics, 23(3), 243-252.

Chaffin, D.B., & Andersson, G.B.J. (1991). Occupational biomechanics (2nd ed.). New York: John Wiley & Sons.

Chen, D.T., & Zeltzer, D. (1992). Pump it up: computer animation of a biomechanically based model of muscle using the finite element method. Computer Graphics, 26, 89-98.

Cheverud, J., Gordon, C.C., Walker, R.A., Jacquish, C., Kohn, L., Moore, A., & Yamashita, N. (1990). 1988 anthropometry survey of U.S. Army personnel: Correlation coefficients and regression equations (Male) (Natick RDEC Technical Report TR-90-033). Natick, Massachusetts: U.S. Army Research, Development, and Engineering Center.

Cooney, W.P., Lucca, M.J., Chao, E.Y.S., & Linsheid, R.L. (1981). The kinesiology of the thumb trapeziometacarpal joint. Journal of Bone and Joint Surgery, 63A, 1371-1381.

Farin, G. (1990). Curves and surfaces for computer aided geometric design. San Diego: Academic Press.

Featherstone, R. (1987). Robot dynamics algorithms. Boston: Kluwer Academic Publishers.

Gordon, C.C., Bradtmiller, B., Churchill, T., Clauser, C.E., McConville, J.T., Tebbetts, I., & Walker, R.A. (1989). 1988 anthropometry survey of U.S. Army personnel: Methods and summary statistics (Natick RDEC Technical Report TR--89-044). Natick: U.S. Army Research, Development, and Engineering Center.

Greiner, T.M. (1991). Hand anthropometry of U.S. Army personnel (Natick RDEC Technical Report TR-92-011). Natick: U.S. Army Research, Development, and Engineering Center.

Grosso, M.R., Quach, R.D., Otani, E., Zhao, J., Wei, S., Ho, P-H, Lu, J., & Badler, N.I. (1989). Anthropometry for computer human figures (Technical Report MS-CIS-89-71). Philadelphia: University of Pennsylvania, Department of Computer and Information Science.

Joseph, J. (1951). Further studies of the metacarpo-phalangeal and interphalangeal joints of the thumb. Journal of Anatomy, 85, 221-229.

Louis, R. (1983). Surgery of the spine. Berlin: Springer-Verlag.

McMinn, R.M.H., & Hutchings, R.T. (1977). A colour atlas of human anatomy. London: Wolfe Medical Publications.

Meindl, R.S., Zehner, G.F., & Hudson, J.A. (1993). A multivariate anthropometric method for crew station design (statistical techniques) (Technical Report AL:CF-TR-1993-0054). Dayton: Wright-Patterson Air Force Base, Aerospace Medical Research Lab.

Monheit, G., & Badler, N. (1991). A kinematic model of the human spine and torso. IEEE computer Graphics and Applications, 11(2), 29-38.

Morrison, D.F. (1967). Multivariate statistical methods. New York: McGraw-Hill.

Muller, M.E. (1959). A comparison of methods for generating normal deviates on digital computers. Journal of the ACM, 6, 376-383.

National Aeronautics and Space Administration (1987). Man-system integration standards (NASA-STD-3000). Houston, TX: NASA Johnson Space Center.

Otani, E. (1989). Software tools for dynamic and kinematic modeling of human motion (Technical Report MS-CIS-89-43). Philadelphia: University of Pennsylvania, Department of Computer and Information Science.

Parry, S.R. (1986). Free-form deformation in a constructive solid geometry modeling system. Unpublished doctoral dissertation, Brigham Young University.

Paul, R.P. (1981). Robot manipulators: Mathematics, programming, and control. Cambridge, MA: MIT Press.

Phillips, C.B. (1991a). Interactive postural control of articulated geometric figures. Unpublished doctoral dissertation, University of Pennsylvania.

Phillips, C.B. (1991b). Jack 5 user's guide (Technical Report MS-CIS-91-78). Philadelphia: University of Pennsylvania, Department of Computer and Information Science.

Phillips, C.B. (1991c). Programming with Jack (Technical Report MS-CIS-91-19). Philadelphia: University of Pennsylvania, Department of Computer and Information Science.

Piegl, L. (1990). On nurbs: A survey. IEEE Computer Graphics and Applications, 11(1), 55-71.

Roche, A.F. (1978). Bone growth and maturation. In F. Falkner & J.M. Tanner (Eds.), Human growth, Volume 2: Postnatal growth (pp. 317-356). New York: Plenum Press.

Roebuck, J.A. Jr. (1995). Anthropometric methods: Designing to fit the human body. Santa Monica, CA: Human Factors and Ergonomics Society.

Sederberg, T.W., & Parry, S.R. (1986). Free-form deformation of solid geometric models. In D.C. Evans & R.J. Athay (Eds.), Computer Graphics: SIGGRAPH '86 proceedings, 20, 151-160.

Tortora, G.J., & Anagnostakos, N.P. (1993). Principles of anatomy and physiology (7th ed.). New York: Harpercollins.

Ulam, S., & Metropolis, N. (1949). The Monte Carlo method. Journal of the American Statistical Association, 44, 335-341.

Vander, A.J., Sherman, J.H., & Luciano, D.S. (1980). Human physiology (3rd ed.). New York: McGraw-Hill.

van der Helm, F.C.T., Veeger, H.E.J., Pronk, G.M., van der Woude, L.H.V., & Rozendal, R.H. (1991). Geometry parameters for musculoskeletal modelling of the shoulder system. Journal of Biomechanics, 25(2), 129-144.

Webb Associates (1978). Anthropomorphic source book, volume 2: A handbook of anthropometric data (NASA Technical Report RP-1024). Houston, TX: National Aeronautics and Space Administration.

Yoshikawa, T. (1990). Foundations of Robotics. Cambridge, MA: MIT Press.

Zhao, J. (1993). Moving posture reconstruction from perspective projections of jointed figure motion. Unpublished doctoral dissertation, University of Pennsylvania.

APPENDIX A

NORMALIZATION AND SCALING OF BODY SEGMENTS

NORMALIZATION AND SCALING OF BODY SEGMENTS

The following is the general algorithm used to prepare the geometry of a given segment for scaling, i.e., segment normalization:

- Given a body segment, find the two cross sections of the segment geometry through the proximal and distal joints (sites) in the x-y plane perpendicular to the z (major) axis of the segment. Compute the x,y bounding rectangle of each the proximal and the distal cross sections. For example, the distal cross section dimensions would be distal_x_min, distal_x_max, distal_y_min, distal_y_max .

- Using the bounding rectangle dimensions, find the scale factors distal_scale_x, distal_scale_y, proximal_scale_x, proximal_scale_y in the x and y directions, which map the rectangles to $-1<x<1$ and $-1<y<1$. Where the scale factor is infinite set it to 1.0 by definition.

- For the z axis, the segment length (distance between proximal and distal joint) must be used. The proximal joint maps to $z = 0$ and the distal joint maps to $z = 1$. The representation of the segment could extend outside the range [0,1] due to possible overlaps.

- The segment has now its proximal and distal cross sections in a standard position. Note that these transformations are associated with the geometry of the segment, and have nothing to do with the anthropometric data thus far.

After normalization, we can apply the anthropometric data to scale the segment, as follows:

- Suppose the following data is available (either from a population or from a given individual): segment length, segment proximal circumference, segment distal circumference. Convert the circumferences to x, y scale factors at the appropriate joint. This can be done by computing the major and minor axes of the ellipse with the arclength equal to the circumference in question, so that the ratio of major to minor (or vice versa, depending on which segment dimension x or y is bigger) axis length is the same as the distal_scale_x to distal_scale_y (proximal_scale_x to proximal_scale_y) ratio at each joint. A simpler option is to assume both x, y are equal.

- Transform the segment geometry according to the length scale factor, that is, new_segment_length = segment_length (scale) * normalized length

- Transform the circumference of the segment by a scale transformation which varies along the z axis of the segment from 0 to new_segment_length, that is, local_segment_scale_x = ((new_segment_distal_scale_x) - (new_segment_proximal_scale_x)) * z + new_segment_ proximal_scale_x. This is a linear scaling in the x direction. The process is similar for the y direction.

APPENDIX B

TORSO DISTRIBUTION FACTORS

TORSO DISTRIBUTION FACTORS

The following are torso distribution factors used for figure creation. These factors have already been incorporated in the formulae for the target anthropometric data representation appearing in Chapter 5.

```
//
//
// Torso: 17 segments distribution
//
//                    width    length    depth
// ------------------------------------------
thorax1   t1      0.8166 0.04309 0.7150
thorax2   t2      0.9477 0.04567 0.8600
thorax3   t3      0.9812 0.04235 0.9178
thorax4   t4      1.0000 0.06077 0.8900
thorax5   t5      0.9693 0.04899 0.9612
thorax6   t6      0.9693 0.05193 0.9677
thorax7   t7      0.9693 0.05193 0.9852
thorax8   t8      0.9693 0.05193 0.9852
thorax9   t9      0.9693 0.05488 0.9900
thorax10 t10      0.9693 0.05967 0.9825
thorax11 t11      0.9442 0.03867 0.9600
thorax12 t12      0.9442 0.08361 0.9350
lumbar1   l1      0.9442 0.07845 0.9000
lumbar2   l2      0.9275 0.07845 0.9000
lumbar3   l3      0.9066 0.05782 0.9000
lumbar4   l4      0.9066 0.07514 0.9000
lumbar5   l5      0.9066 0.07661 0.9300
```

APPENDIX C

POSTURES FOR MEASUREMENT

POSTURES FOR MEASUREMENT

The following figures show all the postures used to perform anthropometric measurements. All these postures follow those described in ANSUR-88. Refer to Chapter 5 for details about which measurements are performed with the different postures.

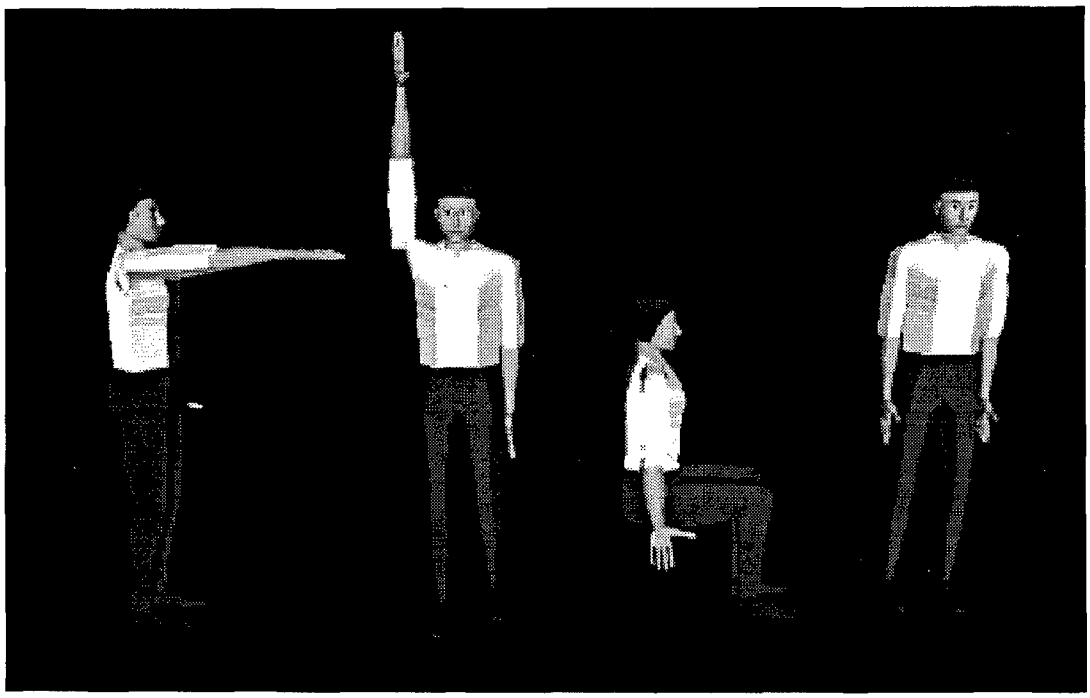

Figure C-1. From left to right: Functional, overhead, sitting, and standing postures.

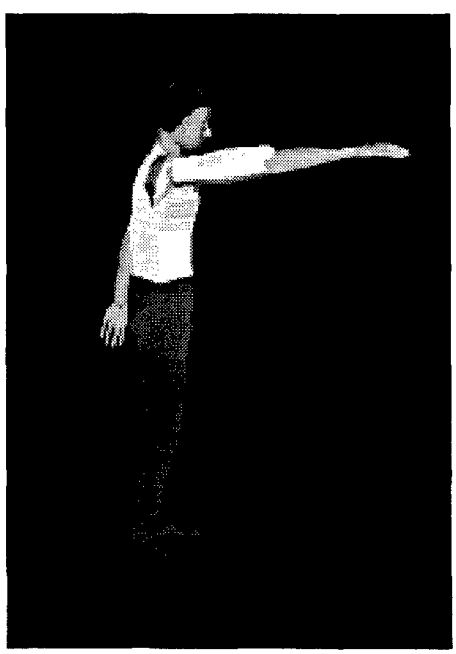

Figure C-2. Functional Extended posture.

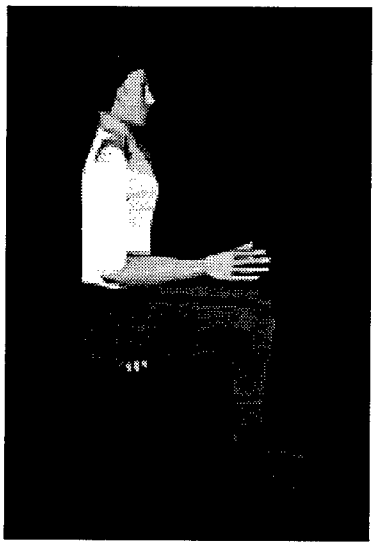

Figure C-3. Sitting forearm up posture.

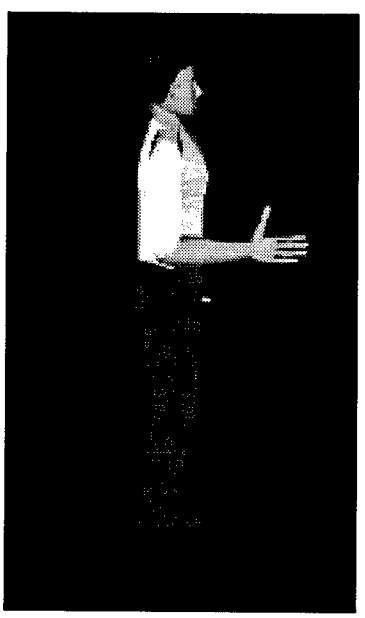

Figure C-4. Standing forearm up posture.

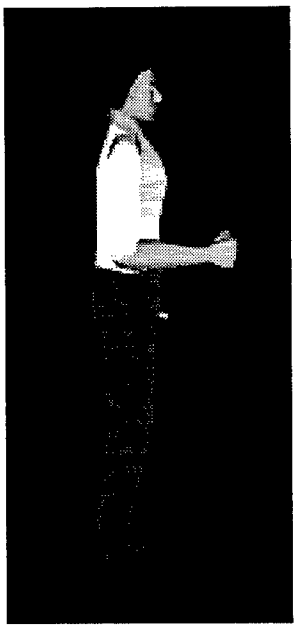

Figure C-5. Forearm up (hand closed) posture.

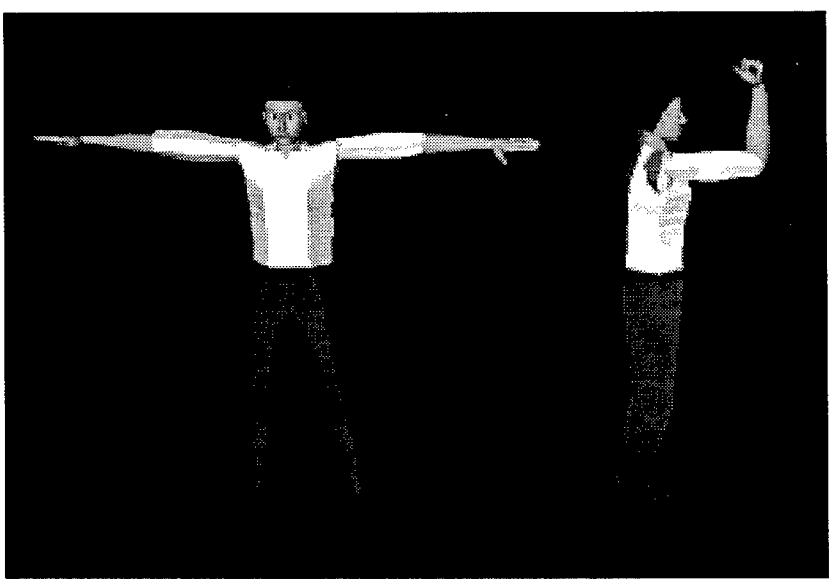

Figure C-6. From left to right: Arm span and biceps-flex postures.

APPENDIX D

NORMALIZATION SITES - LINK SYSTEM

NORMALIZATION SITES - LINK SYSTEM

Table D-1 shows the sites used during normalization of segments. If no sites were involved, it would have meant that the geometry was normalized based on a bounding box approach (i.e., *max - min*). The use of normalization sites assures that the distance between the two sites is a "true" distance. This is necessary to account for segments that may overlap. The distance between sites is effectively scaled when the scaling value is applied. Note that there are only two sites, usually along the longest axis of the segment. For the other two scaling axes, a bounding box approach is always used.

The distance between the two joint center sites represents the link length associated with a given segment. For the torso segments, the link length (i.e., torso length) is spread across the 17 segments.

Table D-1

Normalization Sites

Segment Name	Site 1	Site 2
foot	NS_proximal	NS_distal
toes	*bounding box*	*used*
lower_leg	NS_proximal	NS_distal
upper_leg	NS_proximal	NS_distal
bottom_head	NS_proximal	NS_distal
neck	NS_proximal	NS_distal
clavicle	*bounding box*	*used*
upper_arm	NS_proximal	NS_distal
lower_arm	NS_proximal	NS_distal
t1	NS_proximal	NS_distal
t2	NS_proximal	NS_distal
t3	NS_proximal	NS_distal
t4	NS_proximal	NS_distal
t5	NS_proximal	NS_distal
t6	NS_proximal	NS_distal
t7	NS_proximal	NS_distal
t8	NS_proximal	NS_distal
t9	NS_proximal	NS_distal
t10	NS_proximal	NS_distal
t11	NS_proximal	NS_distal
t12	NS_proximal	NS_distal

Table D-1 (continued)

l1	NS_proximal	NS_distal
l2	NS_proximal	NS_distal
l3	NS_proximal	NS_distal
l4	NS_proximal	NS_distal
l5	NS_proximal	NS_distal
lower_torso	NS_proximal	NS_distal
palm	NS_proximal	NS_distal
finger32	NS_proximal	NS_distal
finger31	NS_proximal	NS_distal
finger30	NS_proximal	NS_distal
finger22	NS_proximal	NS_distal
finger21	NS_proximal	NS_distal
finger20	NS_proximal	NS_distal
finger12	NS_proximal	NS_distal
finger11	NS_proximal	NS_distal
finger10	NS_proximal	NS_distal
finger02	NS_proximal	NS_distal
finger01	NS_proximal	NS_distal
finger00	NS_proximal	NS_distal
thumb2	NS_proximal	NS_distal
thumb1	NS_proximal	NS_distal
thumb0	NS_proximal	NS_distal

APPENDIX E

DEFAULT ANTHROPOMETRIC VARIABLES

DEFAULT ANTHROPOMETRIC VARIABLES

Table E-1 contains a summary of the default anthropometric variables considered by Genfig and Jack.

Table E-1

Default Set of Anthropometric Variables

List #	Variable Name	Variable #
1	ACRHGHT	2
2	ACRHTST	3
3	ACRDLG	4
4	ANKLCIRC	5
5	BLFTLG	9
6	BCRMBDTH	10
7	BICIRCFL	11
8	BIDLBDTH	12
9	BIMBDTH	13
10	BISBDTH	14
11	BIZBDTH	21
12	BUTTDPTH	24
13	BUTTKLTH	26
14	BUTTPLTH	27
15	CALFCIRC	28
16	CERVHT	30
17	CERVSIT	31
18	CHSTBDTH	32
19	CHSTDPTH	36
20	CRCHHGHT	38
21	ELRHGHT	48
22	EYEHTSIT	49
23	FTBRHOR	50
24	FOOTLG	51
25	FCIRCFL	52
26	FORHDLG	54
27	HANDBRTH	57
28	HANDLG	59
29	HEADBRTH	60
30	HEADLGTH	62
31	HIPBRTH	65
32	ILCRSIT	67
33	INPUPBTH	68
34	INSCYE1	69

List #	Variable Name	Variable #
35	KNEECIRC	71
36	KNEEHTPMP	72
37	KNEEHTSIT	73
38	LATFEMEP	74
39	LATMALHT	75
40	NECKCIRC	80
41	OVHDFTRH	83
42	POPHGHT	86
43	RASTL	87
44	SHOUELLT	91
45	SHOULGTH	92
46	SITTHGT	93
47	SPAN	98
48	STATURE	99
49	SUPSTRHT	101
50	TENRIBHT	102
51	THGHCIRC	103
52	THGHCLR	104
53	THMBTPR	106
54	TROCHHT	107
55	WSTBRTH	112
56	WSTDPTH	115
57	WSTHNI	118
58	WSTHSTNI	120
59	WSHTSTOM	121
60	WEIGHT	124
61	WRCTRGRL	125
62	WRISCIRC	126
63	WRINFNGL	129
64	WRTHLGTH	130
65	WRWALLLN	131
66	WRWALLEX	132
67	ECTORBT (H15)	232
68	INFORBB (H22)	239
69	TRAGT (H44)	254

APPENDIX F

THE FIGURE FILE

THE FIGURE FILE

The following is an example of the figure files created by Genfig Version 6.0. The files are structured as follows:

- Declaration and definition of attributes.
- Declaration and definition of segments.
- Declaration and definition of joints.
- Declaration and definition of strength equations.
- Declaration of posture files.
- Declaration of root site.

```
figure {
   attribute eyewhite {
      rgb = (1.00,0.89,0.89);
      ambient = 0.19;
      diffuse = 0.76;
   }
   attribute navy {
      rgb = (1.00,1.00,1.00);
      ambient = 0.00;
      diffuse = 0.01;
   }
   attribute eyes {
      rgb = (1.00,0.49,0.37);
      ambient = 0.09;
      diffuse = 0.34;
   }
   attribute shoes {
      rgb = (1.00,1.00,1.00);
      ambient = 0.04;
      diffuse = 0.16;
   }
   attribute pants {
      rgb = (0.20,0.20,1.00);
      ambient = 0.10;
      diffuse = 0.90;
   }
   attribute flesh {
      rgb = (1.00,0.68,0.57);
```

```
      ambient = 0.20;
      diffuse = 0.80;
    }
    attribute upper_lip {
      rgb = (1.00,0.47,0.47);
      ambient = 0.15;
      diffuse = 0.60;
    }
    attribute hair {
      rgb = (1.00,0.55,0.08);
      ambient = 0.07;
      diffuse = 0.27;
    }
    attribute lower_lip {
      rgb = (1.00,0.52,0.52);
      ambient = 0.17;
      diffuse = 0.70;
    }
    attribute shirt {
      rgb = (1.00,1.00,1.00);
      ambient = 0.20;
      diffuse = 0.80;
    }
    archive = "human-5.9.a";
    segment left_eyeball {
        psurf = "EyeBall66.pss" * scale(2.21,0.73,4.62);
        attribute = (eyewhite,navy,eyes);
        centerofmass = (0.00,0.00,2.31);
        mass = 7.63g;
        site base->location = trans(0.00cm,-1.66cm,2.31cm);
        site sight->location = xyz(90.00deg,0.00deg,90.00deg) * trans(0.00cm,1.01cm,2.31cm);
        site iris->location = trans(0.00cm,-1.66cm,-1.39cm);
        site point->location = trans(0.00cm,-1.66cm,2.42cm);
        site point1->location = xyz(-180.00deg,0.00deg,-180.00deg) * trans(0.00cm,-
1.66cm,3.40cm);
        site LM_pupil_lft->location = trans(0.00cm,1.01cm,2.31cm);
    }
    segment right_eyeball {
        psurf = "EyeBall66.pss" * scale(2.21,0.73,4.62);
        attribute = (eyewhite,navy,eyes);
        centerofmass = (0.00,0.00,2.31);
        mass = 7.63g;
        site base->location = trans(0.00cm,-1.66cm,2.31cm);
        site sight->location = xyz(90.00deg,0.00deg,90.00deg) * trans(0.00cm,1.01cm,2.31cm);
        site iris->location = trans(0.00cm,-1.66cm,-1.39cm);
        site point->location = trans(0.00cm,-1.66cm,2.42cm);
```

```
        site point1->location = xyz(-180.00deg,0.00deg,-180.00deg) * trans(0.00cm,-
1.66cm,3.40cm);
        site LM_pupil_rt->location = trans(0.00cm,0.98cm,2.29cm);
    }
    segment right_foot {
        psurf = "RightFoot66.pss" * scale(19.12,4.97,3.30);
        attribute = shoes;
        centerofmass = (8.41,-1.31,1.65);
        mass = 961.60g;
        site proximal->location = trans(4.25cm,-2.11cm,-1.63cm);
        site distal->location = trans(11.92cm,-2.11cm,5.01cm);
        site normal->location = trans(4.25cm,-2.11cm,5.01cm);
        site new_heel->location = trans(0.01cm,-3.27cm,2.39cm);
        site toes->location = trans(18.89cm,-2.11cm,5.01cm);
        site right->location = trans(17.90cm,4.87cm,3.33cm);
        site left->location = trans(18.84cm,-4.59cm,1.89cm);
        site top->location = trans(1.35cm,4.41cm,-0.99cm);
        site bottom->location = trans(1.32cm,4.40cm,5.01cm);
        site heel->location = trans(0.95cm,-2.11cm,5.01cm);
        site proximal0->location = trans(4.25cm,-2.11cm,-1.63cm);
        site distal0->location = trans(4.25cm,-2.11cm,4.93cm);
        site LM_pternion->location = trans(0.00cm,-3.28cm,2.39cm);
        site LM_heel_pt_rt_lat->location = trans(0.67cm,-0.08cm,5.02cm);
        site LM_heel_pt_rt_med->location = trans(1.52cm,-4.97cm,5.02cm);
        site LM_fifth_metatarsophalangeal_protrusion->location = xyz(0.15deg,0.01deg,-0.43deg)
* trans(18.66cm,4.97cm,3.55cm);
        site LM_first_metatarsophalangeal_protrusion->location = trans(19.00cm,-
4.70cm,1.94cm);
        site LM_bottom_rfoot->location = trans(4.25cm,-2.11cm,5.01cm);
        site LM_heel->location = trans(0.00cm,-3.29cm,2.39cm);
    }
    segment right_toes {
        psurf = "RightToes66.pss" * scale(7.49,4.97,1.10);
        attribute = shoes;
        centerofmass = (3.68,0.24,0.17);
        mass = 106.84g;
        site proximal->location = trans(0.22cm,-2.31cm,0.97cm);
        site distal->location = xyz(-44.36deg,90.00deg,-44.36deg) * trans(1.35cm,-
3.37cm,0.92cm);
        site toetip->location = trans(7.52cm,-2.31cm,0.97cm);
        site LM_acropodion->location = trans(7.48cm,0.31cm,-0.12cm);
    }
    segment left_foot {
        psurf = "LeftFoot66.pss" * scale(19.12,4.97,3.30);
        attribute = shoes;
        centerofmass = (8.41,-1.31,1.65);
```

```
        mass = 961.60g;
        site proximal->location = trans(4.25cm,2.11cm,-1.63cm);
        site distal->location = trans(11.92cm,2.11cm,4.93cm);
        site normal->location = trans(4.25cm,2.11cm,4.93cm);
        site toes->location = trans(18.89cm,2.11cm,4.93cm);
        site new_heel->location = trans(0.00cm,3.27cm,2.31cm);
        site heel->location = trans(0.95cm,2.11cm,4.93cm);
        site proximal0->location = trans(4.25cm,2.11cm,-1.63cm);
        site distal0->location = trans(4.25cm,2.11cm,4.93cm);
        site LM_heel_pt_rt_lat->location = trans(0.67cm,0.08cm,5.02cm);
        site LM_heel_pt_lft_med->location = trans(1.52cm,4.97cm,5.02cm);
        site LM_heel->location = trans(0.00cm,3.29cm,2.39cm);
        site LM_heel_pt_lft_lat->location = trans(0.69cm,0.07cm,5.02cm);
        site LM_pternion->location = trans(0.00cm,3.28cm,2.39cm);
        site LM_fifth_metatarsophalangeal_protrusion->location = trans(18.66cm,-
4.97cm,3.55cm);
        site LM_first_metatarsophalangeal_protrusion->location =
trans(19.00cm,4.70cm,1.94cm);
        site LM_bottom_lfoot->location = trans(4.25cm,2.11cm,5.01cm);
    }
    segment left_toes {
        psurf = "LeftToes66.pss" * scale(7.49,4.97,1.10);
        attribute = shoes;
        centerofmass = (3.68,0.24,0.17);
        mass = 106.84g;
        site proximal->location = trans(0.22cm,2.31cm,0.97cm);
        site distal->location = xyz(-44.36deg,90.00deg,-44.36deg) * trans(1.35cm,3.37cm,0.92cm);
        site toetip->location = trans(7.52cm,2.31cm,0.97cm);
        site LM_acropodion->location = trans(7.48cm,-0.31cm,-0.12cm);
    }
    segment right_lower_leg {
        psurf = "RtLoLeg66.pss" * scale(5.93,5.93,42.67);
        attribute = pants;
        centerofmass = (0.27,0.29,18.78);
        mass = 3.51kg;
        site proximal->location = xyz(0.36deg,-0.81deg,0.00deg) * trans(0.38cm,0.37cm,7.00cm);
        site distal->location = trans(0.00cm,0.00cm,49.68cm);
        site knee->location = trans(5.93cm,0.09cm,7.16cm);
        site back->location = trans(-5.87cm,0.00cm,17.77cm);
        site front->location = trans(5.14cm,0.00cm,16.54cm);
        site right->location = trans(-0.69cm,5.93cm,16.87cm);
        site left->location = trans(-0.74cm,-5.93cm,17.05cm);
        site distal0->location = trans(0.00cm,0.00cm,49.68cm);
        site proximal0->location = xyz(0.36deg,-0.81deg,0.00deg) * trans(0.38cm,0.37cm,7.00cm);
        site LM_back->location = trans(-5.87cm,0.00cm,17.77cm);
        site LM_front->location = trans(5.14cm,0.00cm,16.54cm);
```

```
        site LM_right->location = trans(-0.69cm,5.93cm,16.87cm);
        site LM_left->location = trans(-0.74cm,-5.93cm,17.05cm);
        site LM_calf->location = trans(-0.68cm,5.93cm,16.86cm);
        site LM_midpatella->location = trans(5.62cm,-0.08cm,6.39cm);
        site LM_suprapatella->location = trans(5.78cm,0.13cm,5.19cm);
        site LM_dorsal_juncture->location = trans(0.00cm,0.00cm,0.00cm);
        site LM_lateral_malleolus->location = trans(0.55cm,2.18cm,49.68cm);
        site LM_medial_malleolus->location = trans(0.53cm,-1.90cm,49.68cm);
        site LM_knee_pt_ant->location = trans(5.24cm,0.14cm,4.03cm);
        site LM_ankle_front->location = trans(4.40cm,0.10cm,49.71cm);
        site LM_ankle_back->location = trans(-2.68cm,0.01cm,49.61cm);
        site LM_ankle_left->location = trans(0.53cm,-1.90cm,49.77cm);
        site LM_ankle_right->location = trans(0.55cm,2.18cm,49.72cm);
        site LM_midp_back->location = trans(-4.96cm,0.00cm,8.16cm);
        site LM_midp_left->location = trans(0.83cm,-5.83cm,6.36cm);
        site LM_midp_right->location = trans(1.05cm,5.63cm,6.39cm);
    }
    segment left_lower_leg {
        psurf = "LtLoLeg66.pss" * scale(5.93,5.93,42.67);
        attribute = pants;
        centerofmass = (0.27,0.29,18.78);
        mass = 3.51kg;
        site proximal->location = xyz(-0.36deg,-0.81deg,0.00deg) * trans(0.38cm,-0.37cm,7.00cm);
        site distal->location = trans(0.00cm,0.00cm,49.68cm);
        site distal0->location = trans(0.00cm,0.00cm,49.68cm);
        site proximal0->location = xyz(-0.36deg,-0.81deg,0.00deg) * trans(0.38cm,-
0.37cm,7.00cm);
        site LM_back->location = trans(-5.87cm,0.00cm,17.77cm);
        site LM_front->location = trans(5.14cm,0.00cm,16.54cm);
        site LM_right->location = trans(-0.69cm,-5.93cm,16.87cm);
        site LM_left->location = trans(-0.74cm,5.93cm,17.05cm);
        site LM_calf->location = trans(-0.68cm,-5.93cm,16.86cm);
        site LM_midpatella->location = trans(5.62cm,0.08cm,6.39cm);
        site LM_suprapatella->location = trans(5.78cm,-0.13cm,5.19cm);
        site LM_dorsal_juncture->location = trans(0.00cm,0.00cm,0.00cm);
        site LM_lateral_malleolus->location = trans(0.55cm,-2.18cm,49.68cm);
        site LM_medial_malleolus->location = trans(0.53cm,1.90cm,49.68cm);
        site LM_knee_pt_ant->location = trans(5.24cm,-0.14cm,4.03cm);
        site LM_ankle_front->location = trans(4.40cm,-0.10cm,49.71cm);
        site LM_ankle_back->location = trans(-2.68cm,-0.01cm,49.61cm);
        site LM_ankle_left->location = trans(0.53cm,1.90cm,49.77cm);
        site LM_ankle_right->location = trans(0.55cm,-2.18cm,49.72cm);
        site LM_midp_back->location = trans(-4.96cm,0.00cm,8.16cm);
        site LM_midp_left->location = trans(0.83cm,5.83cm,6.36cm);
        site LM_midp_right->location = trans(1.05cm,-5.63cm,6.39cm);
    }
```

```
    segment right_upper_leg {
        psurf = "RtUpLeg66.pss" * scale(9.36,8.35,42.02);
        attribute = pants;
        centerofmass = (0.69,0.59,17.23);
        mass = 7.63kg;
        site proximal->location = xyz(0.00deg,3.20deg,0.00deg) * trans(-1.17cm,1.30cm,2.05cm);
        site distal->location = trans(0.83cm,1.30cm,44.07cm);
        site back->location = trans(-9.36cm,4.35cm,5.87cm);
        site front->location = trans(9.26cm,4.57cm,5.71cm);
        site poplit->location = trans(-5.88cm,0.71cm,43.49cm);
        site right->location = trans(0.79cm,8.32cm,5.86cm);
        site left->location = trans(0.98cm,-8.35cm,5.88cm);
        site poplit2->location = trans(-5.88cm,0.71cm,38.50cm);
        site proximal0->location = xyz(0.00deg,3.20deg,0.00deg) * trans(-1.66cm,1.30cm,7.79cm);
        site distal0->location = trans(0.83cm,1.30cm,44.07cm);
        site knee->location = trans(8.63cm,1.23cm,44.69cm);
        site LM_back->location = trans(-9.36cm,4.35cm,5.87cm);
        site LM_front->location = trans(9.26cm,4.57cm,5.71cm);
        site LM_right->location = trans(0.79cm,8.32cm,5.86cm);
        site LM_left->location = trans(0.98cm,-8.35cm,5.88cm);
        site LM_dorsal_juncture->location = trans(-7.29cm,1.92cm,35.39cm);
        site LM_lat_femoral_epicondyle_standing->location = trans(-1.18cm,5.19cm,42.29cm);
        site LM_popliteal->location = trans(-5.88cm,0.71cm,43.49cm);
        site LM_thigh_back->location = trans(-9.36cm,4.35cm,5.87cm);
        site LM_thigh_front->location = xyz(-0.04deg,0.13deg,0.05deg) *
trans(9.10cm,4.53cm,5.65cm);
        site LM_popliteal_sit->location = trans(-5.88cm,0.71cm,38.50cm);
    }
    segment left_upper_leg {
        psurf = "LtUpLeg66.pss" * scale(9.36,8.35,42.02);
        attribute = pants;
        centerofmass = (0.69,0.59,17.23);
        mass = 7.63kg;
        site proximal->location = xyz(0.00deg,3.20deg,0.00deg) * trans(-1.17cm,-1.30cm,2.05cm);
        site distal->location = trans(0.83cm,-1.30cm,44.07cm);
        site proximal0->location = xyz(0.00deg,3.20deg,0.00deg) * trans(-1.66cm,-
1.30cm,7.79cm);
        site distal0->location = trans(0.83cm,-1.30cm,44.07cm);
        site LM_back->location = trans(-9.36cm,-4.35cm,5.87cm);
        site LM_front->location = trans(9.26cm,-4.57cm,5.71cm);
        site LM_right->location = trans(0.79cm,-8.32cm,5.86cm);
        site LM_left->location = trans(0.98cm,8.35cm,5.88cm);
        site LM_dorsal_juncture->location = trans(-7.29cm,-1.92cm,35.39cm);
        site LM_lat_femoral_epicondyle_standing->location = trans(-1.18cm,5.19cm,42.29cm);
        site LM_popliteal->location = trans(-5.88cm,-0.71cm,43.49cm);
        site LM_thigh_back->location = trans(-9.36cm,-4.35cm,5.87cm);
```

```
        site LM_thigh_front->location = xyz(-0.04deg,0.13deg,0.05deg) * trans(9.10cm,-
4.53cm,5.65cm);
        site LM_popliteal_sit->location = trans(-5.88cm,-0.71cm,38.50cm);
    }
    segment upper_torso {
        centerofmass = (1.00,0.00,0.00);
        mass = 7.63kg;
        site proximal->location = trans(0.00cm,0.00cm,0.00cm);
        site distal->location = xyz(0.00deg,16.75deg,0.00deg) * trans(2.23cm,-0.04cm,-4.16cm);
        site left->location = xyz(180.00deg,0.00deg,0.00deg) * trans(0.00cm,19.71cm,-3.42cm);
        site right->location = xyz(-180.00deg,0.00deg,0.00deg) * trans(0.00cm,-19.71cm,-3.42cm);
        site lclav->location = xyz(-90.00deg,0.00deg,-5.00deg) * trans(2.50cm,7.57cm,-5.39cm);
        site rclav->location = xyz(90.00deg,0.00deg,5.00deg) * trans(2.50cm,-7.57cm,-5.39cm);
    }
    segment bottom_head {
        psurf = "Head66.pss" * scale(9.74,7.55,12.78);
        attribute = (flesh,upper_lip,hair,lower_lip);
        centerofmass = (-0.17,0.08,10.60);
        mass = 6.03kg;
        site right_eyeball->location = xyz(90.00deg,0.00deg,-90.00deg) * trans(4.49cm,-
2.45cm,10.90cm);
        site left_eyeball->location = xyz(90.00deg,0.00deg,-90.00deg) *
trans(4.49cm,2.45cm,10.90cm);
        site proximal->location = trans(1.93cm,0.00cm,9.79cm);
        site sight->location = xyz(44.96deg,-90.00deg,-44.96deg) * trans(9.12cm,0.00cm,10.90cm);
        site top->location = trans(1.93cm,0.00cm,22.59cm);
        site eye_level->location = trans(-4.14cm,-8.94cm,10.90cm);
        site eye_lvl_out->location = trans(-18.49cm,0.00cm,10.90cm);
        site top_out->location = trans(-18.49cm,0.00cm,22.59cm);
        site bottom->location = trans(-4.14cm,-8.88cm,0.76cm);
        site back->location = trans(-9.74cm,0.01cm,14.19cm);
        site front->location = trans(8.04cm,0.05cm,13.50cm);
        site right->location = trans(1.57cm,-6.22cm,9.56cm);
        site left->location = trans(1.57cm,6.22cm,9.56cm);
        site distal0->location = trans(1.93cm,0.00cm,22.59cm);
        site proximal0->location = trans(1.93cm,0.00cm,9.79cm);
        site top_side->location = trans(-4.62cm,-8.94cm,22.59cm);
        site LM_right->location = trans(1.57cm,-6.22cm,9.56cm);
        site LM_left->location = trans(1.57cm,6.22cm,9.56cm);
        site LM_back->location = trans(-9.74cm,0.01cm,14.19cm);
        site LM_chin->location = trans(7.17cm,0.06cm,1.08cm);
        site LM_glabella->location = trans(8.05cm,0.05cm,13.49cm);
        site LM_frontotemporale_r->location = trans(5.39cm,-5.18cm,12.75cm);
        site LM_frontotemporale_l->location = trans(5.39cm,5.30cm,12.75cm);
        site LM_ectocanthus->location = trans(6.08cm,-4.01cm,10.90cm);
        site LM_alare_r->location = trans(7.74cm,-1.45cm,6.99cm);
```

```
    site LM_alare_l->location = trans(7.74cm,1.45cm,6.99cm);
    site LM_cheilion_r->location = trans(6.60cm,-2.78cm,3.75cm);
    site LM_cheilion_l->location = trans(6.60cm,2.90cm,4.16cm);
    site LM_crinion->location = trans(7.13cm,0.02cm,16.50cm);
    site LM_ear_top->location = trans(1.59cm,-7.05cm,12.17cm);
    site LM_ear_pt->location = trans(-0.15cm,-7.49cm,10.62cm);
    site LM_ear_bottom->location = trans(1.10cm,-6.32cm,6.98cm);
    site LM_infraorbitale_r->location = trans(5.35cm,-2.49cm,8.73cm);
    site LM_infraorbitale_l->location = trans(5.35cm,2.49cm,8.73cm);
    site LM_menton->location = trans(5.72cm,0.10cm,0.00cm);
    site LM_pronasale->location = trans(9.61cm,0.05cm,7.53cm);
    site LM_promenton->location = trans(7.18cm,0.06cm,1.08cm);
    site LM_otobasion_sup->location = trans(2.51cm,-5.95cm,11.13cm);
    site LM_sellion->location = trans(7.67cm,0.05cm,10.63cm);
    site LM_stomion->location = trans(7.70cm,0.05cm,3.81cm);
    site LM_head_top->location = trans(0.68cm,0.08cm,22.48cm);
    site LM_top_of_head->location = trans(-4.17cm,0.10cm,22.56cm);
    site LM_tragion_rt->location = trans(1.90cm,-6.39cm,9.81cm);
    site LM_tragion_lft->location = trans(1.90cm,6.39cm,9.81cm);
    site LM_ectoorbitale_rt->location = trans(4.55cm,-5.45cm,11.72cm);
    site LM_ectoorbitale_lft->location = trans(4.55cm,5.45cm,11.72cm);
    site LM_subnasale->location = trans(7.91cm,0.04cm,6.02cm);
    site LM_gonion_rt->location = trans(0.46cm,-4.97cm,2.71cm);
    site LM_gonion_lft->location = trans(0.46cm,4.97cm,2.71cm);
    site LM_zygion_rt->location = trans(4.98cm,-5.06cm,9.73cm);
    site LM_zygion_lft->location = trans(4.98cm,5.06cm,9.73cm);
    site LM_zygofrontale_rt->location = trans(5.81cm,-4.40cm,13.39cm);
    site LM_zygofrontale_lft->location = trans(5.81cm,4.40cm,13.39cm);
    site LM_TOP_HEAD->location = trans(1.93cm,0.00cm,22.59cm);
    site LM_TOP_HEAD_SIDE->location = trans(-4.62cm,-8.94cm,22.59cm);
    site LM_TOP_HEAD_BACK->location = trans(-18.49cm,0.00cm,22.59cm);
}
segment neck {
    psurf = "Neck66.pss" * scale(6.00,6.00,10.42);
    attribute = flesh;
    centerofmass = (0.24,0.02,8.06);
    mass = 152.63g;
    site proximal->location = xyz(0.00deg,8.03deg,0.00deg) * trans(-0.99cm,-0.03cm,8.30cm);
    site distal->location = trans(3.76cm,-0.27cm,18.72cm);
    site front->location = xyz(0.00deg,-18.64deg,0.00deg) * trans(5.09cm,0.00cm,7.09cm);
    site back->location = trans(-5.42cm,-0.03cm,8.86cm);
    site base_rt->location = trans(-0.96cm,-6.00cm,7.35cm);
    site proximal0->location = trans(-0.99cm,-0.03cm,8.30cm);
    site distal0->location = trans(-0.99cm,-0.03cm,18.72cm);
    site LM_front->location = trans(5.09cm,0.00cm,7.09cm);
    site LM_back->location = trans(-5.53cm,-0.03cm,7.08cm);
```

```
        site LM_cervicale->location = trans(-4.92cm,0.05cm,11.55cm);
        site LM_infrathyroid->location = trans(5.05cm,0.00cm,7.22cm);
        site LM_submandibular->location = trans(3.60cm,-0.22cm,9.02cm);
        site LM_neck_rt_lat->location = trans(4.28cm,-4.16cm,4.48cm);
        site LM_neck_lft_lat->location = trans(4.28cm,4.16cm,4.48cm);
        site LM_trapezius_pt_rt->location = trans(-1.67cm,-4.42cm,9.24cm);
        site LM_trapezius_pt_lft->location = trans(-1.67cm,4.42cm,9.24cm);
        site LM_cervicale0->location = xyz(-0.51deg,-21.26deg,-0.10deg) * trans(-5.71cm,-
0.01cm,3.90cm);
    }
    segment right_clavicle {
        psurf = "RtClavicle66.pss" * scale(5.00,10.00,19.64);
        attribute = shirt;
        centerofmass = (-0.54,-1.82,10.34);
        mass = 2.29kg;
        site proximal->location = trans(0.00cm,4.74cm,6.11cm);
        site lateral->location = xyz(90.00deg,0.00deg,0.00deg) * trans(0.00cm,2.82cm,16.68cm);
        site shoulder_back->location = trans(-10.77cm,3.44cm,17.96cm);
        site top->location = trans(-0.96cm,4.58cm,14.59cm);
        site LM_clavicle_pt_r->location = trans(2.80cm,6.54cm,16.42cm);
        site LM_midshoulder->location = trans(-0.96cm,8.65cm,9.70cm);
        site LM_clavicle_pt_rt->location = trans(-0.95cm,7.53cm,14.59cm);
        site LM_post_scye_rt->location = trans(-2.73cm,-6.24cm,9.46cm);
        site LM_midscye_rt->location = xyz(-180.00deg,0.00deg,-0.02deg) * trans(-
5.03cm,0.05cm,14.36cm);
    }
    segment left_clavicle {
        psurf = "LtClavicle66.pss" * scale(5.00,10.00,19.64);
        attribute = shirt;
        centerofmass = (-0.54,-1.82,10.34);
        mass = 2.29kg;
        site proximal->location = trans(0.00cm,-4.74cm,6.11cm);
        site lateral->location = xyz(-90.00deg,0.00deg,0.00deg) * trans(0.00cm,-2.82cm,16.68cm);
        site LM_clavicle_pt_l->location = trans(2.82cm,-6.54cm,16.39cm);
        site LM_clavicle_pt_lft->location = trans(-0.98cm,-7.53cm,14.59cm);
        site LM_post_scye_lft->location = trans(-2.73cm,6.24cm,9.46cm);
        site LM_midscye_lft->location = xyz(-180.00deg,0.00deg,-0.02deg) * trans(-
5.03cm,0.05cm,14.36cm);
    }
    segment right_upper_arm {
        psurf = "RightUpArm66.pss";
        attribute = shirt;
        centerofmass = (0.00,0.00,0.42);
        mass = 2.14 kg;
        site proximal->location = trans(0.14cm,0.08cm,0.10cm);
        site distal->location = trans(0.00cm,0.08cm,0.87cm);
```

```
        site shoulder_level->location = trans(0.09cm,-1.37cm,0.00cm);
        site shlder_lvl_out->location = xyz(180.00deg,-5.82deg,0.02deg) * trans(-2.96cm,-
2.73cm,0.00cm);
        site deltoid->location = trans(-0.07cm,0.77cm,0.08cm);
        site right->location = trans(-0.01cm,0.94cm,0.52cm);
        site left->location = trans(-0.01cm,-0.94cm,0.52cm);
        site front->location = trans(0.13cm,0.02cm,0.03cm);
        site back->location = trans(-0.92cm,0.54cm,0.48cm);
        site LM_shoulder_level->location = trans(0.09cm,-1.37cm,0.00cm);
        site LM_shlder_lvl_out->location = xyz(180.00deg,-5.82deg,0.02deg) * trans(-2.96cm,-
2.73cm,0.00cm);
        site LM_right->location = trans(-0.01cm,0.94cm,0.52cm);
        site LM_left->location = trans(-0.01cm,-0.94cm,0.52cm);
        site LM_front->location = trans(0.90cm,0.00cm,0.48cm);
        site LM_back->location = trans(-0.92cm,0.54cm,0.48cm);
        site LM_deltoid->location = trans(-0.07cm,0.77cm,0.08cm);
        site proximal0->location = trans(0.01cm,0.08cm,0.10cm);
        site distal0->location = trans(0.00cm,0.08cm,0.87cm);
        site LM_acromion_r->location = trans(-0.09cm,0.09cm,0.00cm);
        site LM_biceps_pt->location = trans(1.13cm,0.54cm,0.55cm);
        site LM_radiale->location = trans(-0.04cm,0.80cm,0.87cm);
        site LM_ant_scye_upper_arm->location = xyz(174.23deg,87.22deg,95.70deg) *
trans(0.75cm,-0.36cm,0.29cm);
        site LM_deltoid_pt_rt->location = trans(-0.04cm,0.79cm,0.22cm);
        site_scale = ((site)proximal0, (site)distal0, (5.36,5.36,33.52), (7.22,7.22,33.52), "z" );
    }
    segment left_upper_arm {
        psurf = "LeftUpArm66.pss";
        attribute = shirt;
        centerofmass = (0.00,0.00,0.42);
        mass = 2.14 kg;
        site proximal->location = trans(0.14cm,-0.08cm,0.10cm);
        site distal->location = trans(0.00cm,-0.08cm,0.87cm);
        site proximal0->location = trans(0.01cm,-0.08cm,0.10cm);
        site distal0->location = trans(0.00cm,-0.08cm,0.87cm);
        site deltoid->location = trans(-0.07cm,-0.76cm,0.08cm);
        site LM_deltoid->location = trans(-0.07cm,-0.76cm,0.08cm);
        site LM_acromion_l->location = trans(-0.09cm,-0.09cm,0.00cm);
        site LM_deltoid_pt_lft->location = trans(-0.04cm,-0.79cm,0.22cm);
        site LM_shoulder_level->location = trans(0.09cm,1.37cm,0.00cm);
        site LM_shlder_lvl_out->location = xyz(180.00deg,-5.82deg,0.02deg) * trans(-
2.96cm,2.73cm,0.00cm);
        site LM_right->location = trans(-0.01cm,-0.94cm,0.52cm);
        site LM_left->location = trans(-0.01cm,0.94cm,0.52cm);
        site LM_front->location = trans(0.13cm,-0.02cm,0.03cm);
        site LM_back->location = trans(-0.92cm,-0.54cm,0.48cm);
```

```
        site LM_biceps_pt->location = trans(1.13cm,-0.54cm,0.55cm);
        site LM_radiale->location = trans(-0.04cm,-0.80cm,0.87cm);
        site LM_ant_scye_upper_arm->location = xyz(174.23deg,87.22deg,95.70deg) *
trans(0.75cm,0.36cm,0.29cm);
        site_scale = ((site)proximal0, (site)distal0, (5.36,5.36,33.52), (7.22,7.22,33.52), "z" );
    }
    segment right_lower_arm {
        psurf = "RightLoArm66.pss";
        attribute = flesh;
        centerofmass = (0.03,0.02,0.36);
        mass = 1.22 kg;
        site proximal->location = trans(0.21cm,0.09cm,0.11cm);
        site distal->location = trans(0.21cm,0.09cm,0.99cm);
        site elbow->location = trans(-0.81cm,0.37cm,0.05cm);
        site right->location = trans(-0.04cm,0.92cm,0.23cm);
        site left->location = trans(0.00cm,-0.88cm,0.23cm);
        site front->location = trans(0.92cm,0.50cm,0.23cm);
        site back->location = trans(-0.94cm,0.46cm,0.23cm);
        site LM_right->location = trans(-0.04cm,0.92cm,0.23cm);
        site LM_left->location = trans(0.00cm,-0.88cm,0.23cm);
        site LM_front->location = trans(0.92cm,0.50cm,0.23cm);
        site LM_back->location = trans(-0.94cm,0.46cm,0.23cm);
        site LM_elbow->location = trans(-0.81cm,0.37cm,0.05cm);
        site proximal0->location = trans(0.21cm,0.09cm,0.11cm);
        site distal0->location = trans(0.21cm,0.09cm,0.99cm);
        site LM_elbow_crease->location = xyz(-0.29deg,6.13deg,-0.03deg) *
trans(0.92cm,0.00cm,0.25cm);
        site LM_stylion->location = trans(0.44cm,-0.15cm,1.00cm);
        site LM_olecranon_ext->location = trans(-0.88cm,0.09cm,0.11cm);
        site LM_olecranon_flex->location = trans(-0.81cm,0.11cm,0.05cm);
        site LM_styl_front->location = trans(0.44cm,0.09cm,1.00cm);
        site LM_styl_back->location = trans(-0.25cm,0.09cm,1.00cm);
        site LM_styl_left->location = trans(0.08cm,-0.40cm,1.00cm);
        site LM_styl_right->location = trans(0.08cm,0.58cm,1.00cm);
        site_scale = ((site)proximal0, (site)distal0, (4.81,4.81,26.65), (6.72,3.84,26.65), "z" );
    }
    segment left_lower_arm {
        psurf = "LeftLoArm66.pss";
        attribute = flesh;
        centerofmass = (0.03,0.02,0.36);
        mass = 1.22 kg;
        site proximal->location = trans(0.21cm,-0.09cm,0.11cm);
        site distal->location = trans(0.21cm,-0.09cm,0.99cm);
        site proximal0->location = trans(0.21cm,-0.09cm,0.11cm);
        site distal0->location = trans(0.21cm,-0.09cm,0.99cm);
        site LM_left->location = trans(0.00cm,0.88cm,0.23cm);
```

```
        site LM_front->location = trans(0.92cm,-0.50cm,0.23cm);
        site LM_back->location = trans(-0.94cm,-0.46cm,0.23cm);
        site LM_elbow->location = trans(-0.81cm,-0.37cm,0.05cm);
        site LM_elbow_crease->location = xyz(-0.29deg,6.13deg,-0.03deg) *
trans(0.92cm,0.00cm,0.25cm);
        site LM_stylion->location = trans(0.44cm,0.15cm,1.00cm);
        site LM_olecranon_ext->location = trans(-0.88cm,-0.09cm,0.11cm);
        site LM_olecranon_flex->location = trans(-0.81cm,-0.11cm,0.05cm);
        site LM_styl_front->location = trans(0.44cm,-0.09cm,1.00cm);
        site LM_styl_back->location = trans(-0.25cm,-0.09cm,1.00cm);
        site LM_styl_left->location = trans(0.08cm,0.40cm,1.00cm);
        site LM_styl_right->location = trans(0.08cm,-0.58cm,1.00cm);
        site_scale = ((site)proximal0, (site)distal0, (4.81,4.81,26.65), (6.72,3.84,26.65), "z" );
    }
    segment t1 {
        psurf = "Spine_T166.pss" * scale(13.11,1.75,8.57);
        attribute = shirt;
        centerofmass = (0.00,0.95,1.79);
        mass = 1.40kg;
        site cy2->location = xyz(-117.16deg,0.00deg,-90.00deg) * trans(0.00cm,1.84cm,-5.39cm);
        site proximal->location = trans(0.00cm,0.00cm,-5.39cm);
        site distal->location = xyz(-109.50deg,0.00deg,-90.00deg) * trans(0.00cm,1.75cm,-
5.39cm);
        site LM_neck_ant->location = trans(0.04cm,1.68cm,4.32cm);
        site LM_suprasternale->location = trans(0.00cm,1.75cm,3.82cm);
    }
    segment t2 {
        psurf = "Spine_T266.pss" * scale(15.19,2.19,10.38);
        attribute = shirt;
        centerofmass = (-0.01,1.32,-1.61);
        mass = 1.40kg;
        site front->location = trans(0.00cm,0.00cm,10.40cm);
        site back->location = trans(0.00cm,0.02cm,-10.37cm);
        site proximal->location = trans(0.00cm,0.00cm,-7.51cm);
        site distal->location = xyz(-1.53deg,0.00deg,0.00deg) * trans(0.00cm,2.19cm,-7.51cm);
    }
    segment t3 {
        psurf = "Spine_T366.pss" * scale(15.67,1.75,11.10);
        attribute = shirt;
        centerofmass = (-0.02,1.10,-0.87);
        mass = 1.40kg;
        site proximal->location = trans(0.00cm,0.00cm,-8.31cm);
        site distal->location = xyz(1.58deg,0.00deg,0.00deg) * trans(0.00cm,1.75cm,-8.31cm);
    }
    segment t4 {
        psurf = "Spine_T466.pss" * scale(15.98,2.63,10.74);
```

```
    attribute = shirt;
    centerofmass = (0.02,2.45,-0.42);
    mass = 1.40kg;
    site interscye_left->location = trans(15.98cm,0.00cm,-0.21cm);
    site interscye_right->location = trans(-15.98cm,0.00cm,-0.19cm);
    site LM_interscye_left->location = trans(15.98cm,0.00cm,-0.21cm);
    site LM_interscye_right->location = trans(-15.98cm,0.00cm,-0.19cm);
    site front->location = trans(0.00cm,-1.56cm,10.83cm);
    site back->location = trans(0.00cm,0.01cm,-10.73cm);
    site proximal->location = trans(0.00cm,0.00cm,-8.17cm);
    site distal->location = xyz(3.64deg,0.00deg,0.00deg) * trans(0.00cm,2.63cm,-8.17cm);
    site LM_back_pt_rt->location = trans(-11.01cm,1.32cm,-5.55cm);
    site LM_back_pt_lft->location = trans(11.01cm,1.32cm,-5.55cm);
}
segment t5 {
    psurf = "Spine_T566.pss" * scale(15.51,2.19,11.58);
    attribute = shirt;
    centerofmass = (-0.06,1.46,-0.57);
    mass = 1.40kg;
    site proximal->location = trans(0.00cm,0.00cm,-8.92cm);
    site distal->location = xyz(3.95deg,0.00deg,0.00deg) * trans(0.00cm,2.19cm,-8.92cm);
}
segment t6 {
    psurf = "Spine_T666.pss" * scale(15.51,2.19,11.70);
    attribute = shirt;
    centerofmass = (0.00,1.50,-0.15);
    mass = 1.40kg;
    site proximal->location = trans(0.00cm,0.00cm,-9.05cm);
    site distal->location = xyz(4.78deg,0.00deg,0.00deg) * trans(0.00cm,2.19cm,-9.05cm);
}
segment t7 {
    psurf = "Spine_T766.pss" * scale(15.51,2.19,11.94);
    attribute = shirt;
    centerofmass = (-0.02,1.57,0.00);
    mass = 1.40kg;
    site front->location = trans(0.00cm,1.21cm,13.35cm);
    site back->location = trans(0.00cm,1.21cm,-11.94cm);
    site right->location = trans(-15.51cm,2.32cm,0.49cm);
    site left->location = trans(15.51cm,2.32cm,0.49cm);
    site proximal->location = trans(0.00cm,0.12cm,-9.14cm);
    site distal->location = xyz(5.26deg,0.00deg,0.00deg) * trans(0.00cm,2.32cm,-9.14cm);
    site LM_chest_right->location = trans(-15.51cm,2.32cm,0.50cm);
    site LM_chest_left->location = trans(15.51cm,2.32cm,0.49cm);
    site LM_chest_back->location = trans(0.00cm,1.21cm,-11.94cm);
    site LM_chest_front->location = trans(0.00cm,1.21cm,11.74cm);
}
```

```
segment t8 {
    psurf = "Spine_T866.pss" * scale(15.51,2.19,11.94);
    attribute = shirt;
    centerofmass = (-0.04,1.64,-0.08);
    mass = 1.40kg;
    site proximal->location = trans(0.00cm,0.20cm,-9.16cm);
    site distal->location = xyz(4.73deg,0.00deg,0.00deg) * trans(0.00cm,2.41cm,-9.16cm);
    site LM_midspine->location = trans(0.00cm,1.30cm,-11.93cm);
    site LM_midspine0->location = trans(0.00cm,0.01cm,-11.91cm);
    site LM_ant_scye_torso->location = trans(-15.51cm,2.39cm,0.32cm);
}
segment t9 {
    psurf = "Spine_T966.pss" * scale(15.51,2.19,11.94);
    attribute = shirt;
    centerofmass = (0.10,1.60,-0.19);
    mass = 1.40kg;
    site proximal->location = trans(0.00cm,0.00cm,-9.18cm);
    site distal->location = xyz(4.58deg,0.00deg,0.00deg) * trans(0.00cm,2.19cm,-9.18cm);
}
segment t10 {
    psurf = "Spine_T1066.pss" * scale(15.51,2.63,11.82);
    attribute = shirt;
    centerofmass = (0.02,1.82,-0.16);
    mass = 1.40kg;
    site proximal->location = trans(0.00cm,0.00cm,-9.02cm);
    site distal->location = xyz(1.58deg,0.00deg,0.00deg) * trans(0.00cm,2.63cm,-9.02cm);
}
segment t11 {
    psurf = "Spine_T1166.pss" * scale(15.03,1.75,11.58);
    attribute = shirt;
    centerofmass = (0.00,1.32,-0.01);
    mass = 1.40kg;
    site proximal->location = trans(0.00cm,0.17cm,-8.66cm);
    site distal->location = trans(0.00cm,1.92cm,-8.66cm);
}
segment t12 {
    psurf = "Spine_T1266.pss" * scale(15.03,3.51,11.34);
    attribute = shirt;
    centerofmass = (0.01,2.57,-0.08);
    mass = 1.40kg;
    site proximal->location = trans(0.00cm,0.21cm,-8.31cm);
    site distal->location = xyz(-3.07deg,0.00deg,0.00deg) * trans(0.00cm,3.72cm,-8.31cm);
}
segment l1 {
    psurf = "Spine_L166.pss" * scale(15.03,3.51,10.86);
    attribute = shirt;
```

```
        centerofmass = (0.00,2.39,0.00);
        mass = 1.40kg;
        site proximal->location = trans(0.00cm,0.00cm,-7.74cm);
        site distal->location = trans(0.00cm,3.51cm,-7.74cm);
    }
    segment l2 {
        psurf = "Spine_L266.pss" * scale(14.87,3.51,10.86);
        attribute = shirt;
        centerofmass = (0.00,2.39,0.00);
        mass = 1.40kg;
        site proximal->location = trans(0.00cm,0.00cm,-7.74cm);
        site distal->location = trans(0.00cm,3.51cm,-7.74cm);
    }
    segment l3 {
        psurf = "Spine_L366.pss" * scale(14.55,2.63,10.86);
        attribute = shirt;
        centerofmass = (0.00,1.78,0.00);
        mass = 1.40kg;
        site proximal->location = trans(0.00cm,0.00cm,-7.74cm);
        site distal->location = trans(0.00cm,2.63cm,-7.74cm);
        site LM_tenth_rib->location = trans(-8.97cm,0.16cm,8.33cm);
    }
    segment l4 {
        psurf = "Spine_L466.pss" * scale(14.55,3.51,10.86);
        attribute = shirt;
        centerofmass = (0.00,2.38,0.00);
        mass = 1.40kg;
        site proximal->location = trans(0.00cm,0.00cm,-7.74cm);
        site distal->location = trans(0.00cm,3.51cm,-7.74cm);
        site LM_waist_rt->location = trans(-14.55cm,2.09cm,0.00cm);
        site LM_waist_lft->location = trans(14.55cm,2.09cm,0.00cm);
        site LM_waist_ant_navel->location = trans(0.00cm,0.09cm,10.86cm);
        site LM_waist_post->location = trans(-0.06cm,2.09cm,-9.81cm);
    }
    segment l5 {
        psurf = "Spine_L566.pss" * scale(14.55,3.51,11.22);
        attribute = shirt;
        centerofmass = (0.05,2.41,-0.51);
        mass = 1.40kg;
        site right->location = trans(-14.13cm,3.68cm,-0.86cm);
        site left->location = trans(14.17cm,3.69cm,-0.85cm);
        site front->location = trans(-7.07cm,3.68cm,9.12cm);
        site back->location = trans(-7.05cm,3.68cm,-10.86cm);
        site proximal->location = xyz(-90.00deg,0.00deg,-90.00deg) * trans(0.01cm,0.17cm,-
7.98cm);
        site distal->location = trans(0.01cm,3.68cm,-7.97cm);
```

```
        site LM_waist_lft->location = trans(14.34cm,3.68cm,-0.38cm);
        site LM_waist_ant->location = trans(0.01cm,3.68cm,10.28cm);
        site LM_waist_rt->location = trans(-14.33cm,3.68cm,-0.38cm);
        site LM_iliocristale->location = trans(-12.99cm,0.39cm,3.56cm);
    }
    segment lower_torso {
        psurf = "LowerTorso66.pss" * scale(12.32,16.89,14.11);
        attribute = pants;
        centerofmass = (6.16,8.44,12.69);
        mass = 9.62kg;
        site proximal->location = trans(2.72cm,0.00cm,2.92cm);
        site distal->location = trans(-7.67cm,0.00cm,24.61cm);
        site floor->location = xyz(42.08deg,90.00deg,42.08deg) * trans(2.72cm,0.00cm,-
103.13cm);
        site ldistal->location = xyz(-90.00deg,0.00deg,0.00deg) * trans(2.72cm,0.00cm,2.92cm);
        site rdistal->location = xyz(90.00deg,0.00deg,0.00deg) * trans(2.72cm,0.00cm,2.92cm);
        site seat->location = trans(2.72cm,0.00cm,-9.53cm);
        site rhip_lateral->location = xyz(-180.00deg,0.00deg,0.00deg) * trans(0.00cm,-
9.46cm,10.50cm);
        site lhip_lateral->location = xyz(-180.00deg,0.00deg,0.00deg) *
trans(0.00cm,9.46cm,10.50cm);
        site center_of_mass->location = trans(1.77cm,1.13cm,24.56cm);
        site left_hiphandle->location = xyz(108.21deg,-20.58deg,-102.00deg) *
trans(0.73cm,17.57cm,20.17cm);
        site right_hiphandle->location = xyz(-108.21deg,-20.58deg,-67.00deg) * trans(-0.39cm,-
18.57cm,21.37cm);
        site butt->location = trans(-12.57cm,-12.43cm,8.42cm);
        site crotch_level->location = trans(2.71cm,-10.05cm,2.92cm);
        site left_side->location = trans(2.11cm,16.89cm,12.23cm);
        site right_side->location = trans(2.10cm,-16.89cm,12.23cm);
        site sit_ext->location = trans(-22.24cm,0.00cm,4.84cm);
        site sit_level->location = trans(-22.24cm,0.00cm,4.84cm);
        site front->location = trans(11.08cm,-12.42cm,8.43cm);
        site proximal0->location = trans(2.71cm,0.00cm,10.50cm);
        site distal0->location = xyz(0.00deg,90.00deg,0.00deg) * trans(2.73cm,0.00cm,24.61cm);
        site LM_hip_joint_rt->location = trans(0.00cm,-9.46cm,10.50cm);
        site LM_hip_joint_lft->location = trans(0.00cm,9.46cm,10.50cm);
        site LM_buttock_pt_post->location = trans(-12.20cm,0.00cm,13.28cm);
        site LM_buttock_pt_rt_lat->location = trans(-3.10cm,-15.25cm,11.07cm);
        site LM_buttock_pt_lft_lat->location = trans(-3.10cm,15.25cm,11.07cm);
        site LM_gluteal_furrow_pt2_rt->location = trans(-8.58cm,-8.88cm,2.99cm);
        site LM_gluteal_furrow_pt2_lft->location = trans(-8.58cm,8.88cm,2.99cm);
        site LM_ant_sup_iliac_spine_rt->location = trans(11.57cm,-11.58cm,15.72cm);
        site LM_ant_sup_iliac_spine_lft->location = trans(11.57cm,11.58cm,15.72cm);
        site LM_post_iliac_spine_rt->location = trans(-11.67cm,-3.40cm,21.33cm);
        site LM_post_iliac_spine_lft->location = trans(-11.67cm,3.40cm,21.33cm);
```

```
        site LM_trochanter_rt->location = trans(0.07cm,-16.14cm,9.72cm);
        site LM_trochanterion_rt->location = trans(2.59cm,-16.77cm,7.22cm);
        site LM_trochanter_lft->location = trans(0.07cm,16.14cm,9.72cm);
        site LM_trochanterion_lft->location = trans(2.59cm,16.77cm,7.22cm);
        site LM_front_pelvis->location = trans(11.61cm,0.01cm,13.28cm);
        site LM_crotch_level_rt->location = trans(2.71cm,-10.05cm,2.92cm);
        site LM_crotch_level_lft->location = trans(2.71cm,10.05cm,2.92cm);
        site LM_crotch->location = trans(2.71cm,0.22cm,2.92cm);
        site LM_sit_level->location = trans(-22.24cm,0.00cm,4.84cm);
    }
    segment left_palm {
        psurf = "lpalm.pss" * scale(4.48,11.51,1.83);
        attribute = flesh;
        centerofmass = (0.52,5.07,0.08);
        mass = 305.27g;
        site base->location = xyz(-90.00deg,6.56deg,-180.00deg) * trans(-1.52cm,0.00cm,0.02cm);
        site f11->location = xyz(177.02deg,-0.14deg,-175.13deg) * trans(-
3.31cm,11.99cm,0.01cm);
        site f22->location = xyz(177.02deg,-0.14deg,-175.13deg) * trans(-0.98cm,11.51cm,-
0.07cm);
        site f33->location = xyz(177.02deg,-0.14deg,-175.13deg) * trans(1.49cm,11.09cm,0.22cm);
        site f44->location = xyz(177.02deg,-0.14deg,-175.13deg) * trans(3.69cm,9.97cm,0.53cm);
        site left_bird->location = xyz(-94.00deg,6.39deg,-179.88deg) * trans(0.01cm,10.01cm,-
0.77cm);
        site palmcenter->location = trans(0.18cm,7.86cm,-0.09cm);
        site thumb0->location = xyz(-2.98deg,0.14deg,15.13deg) * trans(-1.74cm,2.67cm,0.69cm);
        site LM_hand_closed_center->location = trans(-0.18cm,10.58cm,-0.09cm);
    }
    segment left_finger32 {
        psurf = "lpinfinger02.pss" * scale(1.12,2.56,1.12);
        attribute = flesh;
        centerofmass = (0.03,1.80,-0.06);
        mass = 10.18g;
        site base0->location = trans(0.00cm,0.64cm,-0.11cm);
        site tip->location = trans(0.09cm,3.20cm,0.63cm);
    }
    segment left_finger31 {
        psurf = "lpinfinger01.pss";
        attribute = flesh;
        centerofmass = (-0.05,0.61,-0.01);
        mass = 10.18 g;
        site base0->location = trans(-0.08cm,0.31cm,-0.04cm);
        site tip->location = trans(-0.08cm,1.25cm,-0.01cm);
        site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.34,2.56,1.34), "y" );
    }
    segment left_finger30 {
```

```
    psurf = "lpinfinger00.pss";
    attribute = flesh;
    centerofmass = (-0.12,0.86,0.14);
    mass = 10.18 g;
    site base0->location = trans(-0.30cm,0.26cm,-0.13cm);
    site tip->location = trans(-0.30cm,1.20cm,-0.13cm);
    site LM_metacarpale_V->location = trans(-0.98cm,0.23cm,-0.17cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.79,2.56,1.79), "y" );
}
segment left_finger22 {
    psurf = "lringfinger02.pss" * scale(1.12,2.56,1.12);
    attribute = flesh;
    centerofmass = (0.02,1.78,-0.14);
    mass = 10.18g;
    site base0->location = trans(-0.01cm,0.52cm,0.12cm);
    site tip->location = trans(0.05cm,3.08cm,0.61cm);
}
segment left_finger21 {
    psurf = "lringfinger01.pss";
    attribute = flesh;
    centerofmass = (0.04,0.51,0.13);
    mass = 10.18 g;
    site base0->location = trans(0.02cm,0.22cm,-0.08cm);
    site tip->location = trans(0.01cm,1.08cm,-0.09cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.34,2.56,1.34), "y" );
}
segment left_finger20 {
    psurf = "lringfinger00.pss";
    attribute = flesh;
    centerofmass = (-0.12,0.82,0.10);
    mass = 10.18 g;
    site base0->location = trans(-0.27cm,0.25cm,-0.14cm);
    site tip->location = trans(-0.27cm,1.11cm,-0.14cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.79,2.56,1.79), "y" );
}
segment left_finger12 {
    psurf = "lmidfinger02.pss" * scale(1.12,2.56,1.12);
    attribute = flesh;
    centerofmass = (0.06,1.83,-0.01);
    mass = 10.18g;
    site base0->location = trans(0.07cm,0.69cm,-0.07cm);
    site tip->location = trans(0.08cm,3.25cm,0.44cm);
    site LM_dactylion_III_l->location = trans(0.01cm,3.04cm,0.90cm);
}
segment left_finger11 {
    psurf = "lmidfinger01.pss";
```

```
    attribute = flesh;
    centerofmass = (-0.07,0.58,-0.01);
    mass = 10.18 g;
    site base0->location = trans(-0.06cm,0.22cm,-0.05cm);
    site tip->location = trans(-0.08cm,1.12cm,-0.04cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.34,2.56,1.34), "y" );
}
segment left_finger10 {
    psurf = "lmidfinger00.pss";
    attribute = flesh;
    centerofmass = (0.06,0.75,0.05);
    mass = 10.18 g;
    site base0->location = trans(0.03cm,0.23cm,0.00cm);
    site tip->location = trans(0.05cm,1.14cm,0.01cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.79,2.56,1.79), "y" );
}
segment left_finger02 {
    psurf = "linfinger02.pss" * scale(1.12,2.56,1.12);
    attribute = flesh;
    centerofmass = (-0.09,1.84,-0.16);
    mass = 10.18g;
    site base0->location = trans(-0.02cm,0.67cm,0.00cm);
    site tip->location = trans(-0.11cm,3.22cm,0.58cm);
    site LM_dactylion_II->location = trans(-0.18cm,3.02cm,1.02cm);
}
segment left_finger01 {
    psurf = "linfinger01.pss";
    attribute = flesh;
    centerofmass = (0.00,0.56,-0.10);
    mass = 10.18 g;
    site base0->location = trans(0.15cm,0.24cm,0.04cm);
    site tip->location = trans(0.16cm,1.14cm,0.05cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.34,2.56,1.34), "y" );
}
segment left_finger00 {
    psurf = "linfinger00.pss";
    attribute = flesh;
    centerofmass = (0.22,1.01,-0.09);
    mass = 10.18 g;
    site base0->location = trans(0.38cm,0.23cm,0.21cm);
    site tip->location = trans(0.38cm,1.13cm,0.22cm);
    site LM_metacarpale_II->location = trans(1.12cm,0.24cm,0.21cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.79,2.56,1.79), "y" );
}
segment left_thumb2 {
    psurf = "lthumb02.pss" * scale(1.12,2.56,1.12);
```

```
    attribute = flesh;
    centerofmass = (-0.01,2.37,0.09);
    mass = 10.18g;
    site base0->location = trans(0.00cm,0.76cm,0.07cm);
    site tip->location = trans(-0.30cm,3.32cm,-0.72cm);
    site LM_thumbtip->location = trans(-0.16cm,3.31cm,-0.78cm);
}
segment left_thumb1 {
    psurf = "lthumb01.pss";
    attribute = flesh;
    centerofmass = (0.03,0.36,-0.08);
    mass = 10.18 g;
    site base0->location = trans(0.00cm,0.19cm,0.10cm);
    site tip->location = trans(0.02cm,0.93cm,0.10cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.46,2.56,1.46), "y" );
}
segment left_thumb0 {
    psurf = "lthumb00.pss";
    attribute = flesh;
    centerofmass = (-0.10,0.47,-0.01);
    mass = 10.18 g;
    site tip->location = trans(-0.13cm,0.94cm,0.04cm);
    site base0->location = xyz(0.00deg,0.00deg,-49.75deg) * trans(-0.13cm,0.19cm,0.06cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (2.58,2.56,2.24), "y" );
}
segment right_thumb2 {
    psurf = "rthumb02.pss" * scale(1.12,2.56,1.12);
    attribute = flesh;
    centerofmass = (-0.01,2.37,0.09);
    mass = 10.18g;
    site base0->location = trans(0.00cm,0.76cm,-0.07cm);
    site tip->location = trans(-0.30cm,3.32cm,0.72cm);
    site LM_thumbtip->location = trans(-0.16cm,3.31cm,0.78cm);
}
segment right_thumb1 {
    psurf = "rthumb01.pss";
    attribute = flesh;
    centerofmass = (0.03,0.36,-0.08);
    mass = 10.18 g;
    site base0->location = trans(0.00cm,0.19cm,-0.10cm);
    site tip->location = trans(0.02cm,0.93cm,-0.10cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.46,2.56,1.46), "y" );
}
segment right_thumb0 {
    psurf = "rthumb00.pss";
    attribute = flesh;
```

```
    centerofmass = (-0.10,0.47,-0.01);
    mass = 10.18 g;
    site tip->location = trans(-0.13cm,0.94cm,-0.04cm);
    site base0->location = xyz(0.00deg,0.00deg,-49.75deg) * trans(-0.13cm,0.19cm,-0.06cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (2.58,2.56,2.24), "y" );
}
segment right_finger32 {
    psurf = "rpinfinger02.pss" * scale(1.12,2.56,1.12);
    attribute = flesh;
    centerofmass = (0.03,1.80,-0.06);
    mass = 10.18g;
    site base0->location = trans(0.00cm,0.64cm,0.11cm);
    site tip->location = trans(0.09cm,3.20cm,-0.63cm);
}
segment right_finger31 {
    psurf = "rpinfinger01.pss";
    attribute = flesh;
    centerofmass = (-0.05,0.61,-0.01);
    mass = 10.18 g;
    site base0->location = trans(-0.08cm,0.31cm,0.04cm);
    site tip->location = trans(-0.08cm,1.25cm,0.01cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.34,2.56,1.34), "y" );
}
segment right_finger30 {
    psurf = "rpinfinger00.pss";
    attribute = flesh;
    centerofmass = (-0.12,0.86,0.14);
    mass = 10.18 g;
    site base0->location = trans(-0.30cm,0.26cm,0.13cm);
    site tip->location = trans(-0.30cm,1.20cm,0.13cm);
    site LM_metacarpale_V->location = trans(-0.98cm,0.23cm,0.17cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.79,2.56,1.79), "y" );
}
segment right_finger22 {
    psurf = "rringfinger02.pss" * scale(1.12,2.56,1.12);
    attribute = flesh;
    centerofmass = (0.02,1.78,-0.14);
    mass = 10.18g;
    site base0->location = trans(-0.01cm,0.52cm,-0.12cm);
    site tip->location = trans(0.05cm,3.08cm,-0.61cm);
}
segment right_finger21 {
    psurf = "rringfinger01.pss";
    attribute = flesh;
    centerofmass = (0.04,0.51,0.13);
    mass = 10.18 g;
```

```
    site base0->location = trans(0.02cm,0.22cm,0.08cm);
    site tip->location = trans(0.01cm,1.08cm,0.09cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.34,2.56,1.34), "y" );
}
segment right_finger20 {
    psurf = "rringfinger00.pss";
    attribute = flesh;
    centerofmass = (-0.12,0.82,0.10);
    mass = 10.18 g;
    site base0->location = trans(-0.27cm,0.25cm,0.14cm);
    site tip->location = trans(-0.27cm,1.11cm,0.14cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.79,2.56,1.79), "y" );
}
segment right_finger12 {
    psurf = "rmidfinger02.pss" * scale(1.12,2.56,1.12);
    attribute = flesh;
    centerofmass = (0.06,1.83,-0.01);
    mass = 10.18g;
    site base0->location = trans(0.07cm,0.69cm,0.07cm);
    site tip->location = trans(0.08cm,3.25cm,-0.44cm);
    site LM_dactylion_III_r->location = trans(0.00cm,3.05cm,-0.89cm);
}
segment right_finger11 {
    psurf = "rmidfinger01.pss";
    attribute = flesh;
    centerofmass = (-0.07,0.58,-0.01);
    mass = 10.18 g;
    site base0->location = trans(-0.06cm,0.22cm,0.05cm);
    site tip->location = trans(-0.08cm,1.12cm,0.04cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.34,2.56,1.34), "y" );
}
segment right_finger10 {
    psurf = "rmidfinger00.pss";
    attribute = flesh;
    centerofmass = (0.06,0.75,0.05);
    mass = 10.18 g;
    site base0->location = trans(0.03cm,0.23cm,0.00cm);
    site tip->location = trans(0.05cm,1.14cm,-0.01cm);
    site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.79,2.56,1.79), "y" );
}
segment right_finger02 {
    psurf = "rinfinger02.pss" * scale(1.12,2.56,1.12);
    attribute = flesh;
    centerofmass = (-0.09,1.84,-0.16);
    mass = 10.18g;
    site base0->location = trans(-0.02cm,0.67cm,0.00cm);
```

```
        site tip->location = trans(-0.11cm,3.22cm,-0.58cm);
        site LM_dactylion_II->location = trans(-0.18cm,3.02cm,-1.02cm);
    }
    segment right_finger01 {
        psurf = "rinfinger01.pss";
        attribute = flesh;
        centerofmass = (0.00,0.56,-0.10);
        mass = 10.18 g;
        site base0->location = trans(0.15cm,0.24cm,-0.04cm);
        site tip->location = trans(0.16cm,1.14cm,-0.05cm);
        site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.34,2.56,1.34), "y" );
    }
    segment right_finger00 {
        psurf = "rinfinger00.pss";
        attribute = flesh;
        centerofmass = (0.22,1.01,-0.09);
        mass = 10.18 g;
        site base0->location = trans(0.38cm,0.23cm,-0.21cm);
        site tip->location = trans(0.38cm,1.13cm,-0.22cm);
        site LM_metacarpale_II->location = trans(1.12cm,0.24cm,-0.21cm);
        site_scale = ((site)base0, (site)tip, (1.12,2.56,1.12), (1.79,2.56,1.79), "y" );
    }
    segment right_palm {
        psurf = "rpalm.pss" * scale(4.48,11.51,1.83);
        attribute = flesh;
        centerofmass = (0.52,5.07,0.08);
        mass = 305.27g;
        site base->location = xyz(-90.00deg,-6.56deg,0.00deg) * trans(1.52cm,0.00cm,0.02cm);
        site real_base->location = xyz(-90.00deg,-6.56deg,0.00deg) *
trans(1.52cm,0.82cm,0.02cm);
        site f11->location = xyz(2.98deg,0.14deg,4.87deg) * trans(3.31cm,11.99cm,0.01cm);
        site f22->location = xyz(2.98deg,0.14deg,4.87deg) * trans(0.98cm,11.51cm,-0.07cm);
        site f33->location = xyz(2.98deg,0.14deg,4.87deg) * trans(-1.49cm,11.09cm,0.22cm);
        site f44->location = xyz(2.98deg,0.14deg,4.87deg) * trans(-3.69cm,9.97cm,0.53cm);
        site right_bird->location = xyz(-86.00deg,-6.39deg,0.12deg) * trans(0.01cm,10.01cm,-
0.77cm);
        site palmcenter->location = trans(0.18cm,7.86cm,-0.09cm);
        site thumb0->location = xyz(-177.02deg,-0.14deg,-164.87deg) *
trans(1.74cm,2.67cm,0.69cm);
        site left->location = trans(-4.47cm,9.90cm,0.49cm);
        site right->location = trans(4.48cm,9.90cm,0.49cm);
        site front->location = trans(-0.05cm,7.56cm,-1.61cm);
        site back->location = trans(-0.05cm,7.56cm,1.70cm);
        site LM_hand_closed_center->location = trans(0.18cm,10.58cm,-0.09cm);
    }
    joint rthumb1 {
```

```
    connect right_thumb0.tip to right_thumb1.base0;
    type = R(x);
    llimit = (-45.00deg);
    ulimit = (5.00deg);
}
joint rthumb2 {
    connect right_thumb1.tip to right_thumb2.base0;
    type = R(x);
    llimit = (-75.00deg);
    ulimit = (10.00deg);
}
joint rpinfinger31 {
    connect right_finger30.tip to right_finger31.base0;
    type = R(x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint rpinfinger32 {
    connect right_finger31.tip to right_finger32.base0;
    type = R(x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint rringfinger22 {
    connect right_finger21.tip to right_finger22.base0;
    type = R(x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint rringfinger21 {
    connect right_finger20.tip to right_finger21.base0;
    type = R(x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint rmidfinger12 {
    connect right_finger11.tip to right_finger12.base0;
    type = R(x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint rmidfinger11 {
    connect right_finger10.tip to right_finger11.base0;
    type = R(x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
```

```
}
joint rinfinger02 {
    connect right_finger01.tip to right_finger02.base0;
    type = R(x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint rinfinger01 {
    connect right_finger00.tip to right_finger01.base0;
    type = R(x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint right_finger00 {
    connect right_palm.f11 to right_finger00.base0;
    type = R(z) * R(x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint right_finger10 {
    connect right_palm.f22 to right_finger10.base0;
    type = R(z) * R(x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint right_finger20 {
    connect right_palm.f33 to right_finger20.base0;
    type = R(z) * R(x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint right_finger30 {
    connect right_palm.f44 to right_finger30.base0;
    type = R(z) * R(x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint rthumb0 {
    connect right_palm.thumb0 to right_thumb0.base0;
    type = R(-z) * R(-y);
    llimit = (0.00deg,0.00deg);
    ulimit = (40.00deg,110.00deg);
}
joint right_toes {
    connect right_foot.toes to right_toes.proximal;
    type = R(y);
```

```
    llimit = (0.00deg);
    ulimit = (90.00deg);
}
joint left_toes {
    connect left_foot.toes to left_toes.proximal;
     type = R(y);
    llimit = (0.00deg);
    ulimit = (90.00deg);
}
joint right_ankle {
    connect right_lower_leg.distal to right_foot.proximal;
     type = R(-z) * R(-x) * R(y);
    llimit = (-55.00deg,-39.00deg,-79.60deg);
    ulimit = (63.00deg,35.00deg,25.00deg);
}
joint left_ankle {
    connect left_lower_leg.distal to left_foot.proximal;
     type = R(z) * R(x) * R(y);
    llimit = (-55.00deg,-39.00deg,-79.60deg);
    ulimit = (63.00deg,35.00deg,25.00deg);
}
joint right_knee {
    connect right_upper_leg.distal to right_lower_leg.proximal;
     type = R(-y);
    llimit = (0.00deg);
    ulimit = (160.20deg);
    rest = (10.00deg);
    tolerance = (10.00deg);
}
joint left_knee {
    connect left_upper_leg.distal to left_lower_leg.proximal;
     type = R(-y);
    llimit = (0.00deg);
    ulimit = (160.20deg);
    rest = (10.00deg);
    tolerance = (10.00deg);
}
joint right_hip {
    connect lower_torso.rhip_lateral to right_upper_leg.proximal;
     type = R(-z) * R(-x) * R(y);
    llimit = (-50.00deg,-10.00deg,-17.00deg);
    ulimit = (40.00deg,30.00deg,117.00deg);
}
joint left_hip {
    connect lower_torso.lhip_lateral to left_upper_leg.proximal;
     type = R(z) * R(x) * R(y);
```

```
    llimit = (-50.00deg,-10.00deg,-17.00deg);
    ulimit = (40.00deg,30.00deg,117.00deg);
}
joint atlanto_occipital {
    connect neck.distal to bottom_head.proximal;
    type = R(z) * R(y) * R(x);
    llimit = (-43.50deg,-51.50deg,-23.50deg);
    ulimit = (43.50deg,25.80deg,23.50deg);
}
joint base_of_neck {
    connect t1.distal to neck.proximal;
    type = R(y) * R(z) * R(x);
    llimit = (-51.50deg,-56.10deg,-20.00deg);
    ulimit = (45.20deg,56.10deg,20.00deg);
}
joint spinet2t1 {
    connect t2.distal to t1.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet3t2 {
    connect t3.distal to t2.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet4t3 {
    connect t4.distal to t3.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet5t4 {
    connect t5.distal to t4.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet6t5 {
    connect t6.distal to t5.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet7t6 {
```

```
    connect t7.distal to t6.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-4.00deg,-1.67deg);
    ulimit = (2.50deg,4.00deg,1.67deg);
}
joint spinet8t7 {
    connect t8.distal to t7.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-2.00deg,-1.67deg);
    ulimit = (2.50deg,2.00deg,1.67deg);
}
joint spinet9t8 {
    connect t9.distal to t8.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-2.00deg,-1.67deg);
    ulimit = (2.50deg,2.00deg,1.67deg);
}
joint spinet10t9 {
    connect t10.distal to t9.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-2.00deg,-1.67deg);
    ulimit = (2.50deg,2.00deg,1.67deg);
}
joint spinet11t10 {
    connect t11.distal to t10.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet12t11 {
    connect t12.distal to t11.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-2.40deg,-2.00deg,-1.67deg);
    ulimit = (3.00deg,2.00deg,1.67deg);
}
joint spinel1t12 {
    connect l1.distal to t12.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-3.30deg,-5.00deg,-1.67deg);
    ulimit = (4.00deg,5.00deg,1.67deg);
}
joint spinel2l1 {
    connect l2.distal to l1.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-3.97deg,-1.00deg,-4.00deg);
    ulimit = (7.03deg,1.00deg,4.00deg);
```

```
}
joint spinel3l2 {
    connect l3.distal to l2.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-4.33deg,-1.50deg,-4.00deg);
    ulimit = (7.67deg,1.50deg,4.00deg);
}
joint spinel4l3 {
    connect l4.distal to l3.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-6.50deg,-1.50deg,-4.00deg);
    ulimit = (11.50deg,1.50deg,4.00deg);
}
joint spinel5l4 {
    connect l5.distal to l4.proximal;
     type = R(x) * R(y) * R(z);
    llimit = (-8.66deg,-2.00deg,-4.00deg);
    ulimit = (15.34deg,2.00deg,4.00deg);
}
joint waist {
    connect lower_torso.distal to l5.proximal;
     type = R(y) * R(z) * R(x);
    llimit = (-6.50deg,-2.00deg,-4.00deg);
    ulimit = (11.00deg,2.00deg,4.00deg);
}
joint solar_plexus {
    connect t1.distal to upper_torso.proximal;
    displacement = trans(0.00cm,0.00cm, 0.00cm);
}
joint right_clavicle_joint {
    connect upper_torso.rclav to right_clavicle.proximal;
     type = R(-x) * R(y);
    llimit = (-12.00deg,-8.00deg);
    ulimit = (25.00deg,40.00deg);
}
joint right_shoulder {
    connect right_clavicle.lateral to right_upper_arm.proximal;
     type = R(-z) * R(-x) * R(y);
    llimit = (-108.55deg,-48.00deg,-61.45deg);
    ulimit = (71.50deg,180.95deg,187.65deg);
}
joint right_elbow {
    connect right_upper_arm.distal to right_lower_arm.proximal;
     type = R(y);
    llimit = (0.00deg);
    ulimit = (149.75deg);
```

```
    rest = (5.00deg);
    tolerance = (5.00deg);
}
joint right_wrist {
    connect right_lower_arm.distal to right_palm.base;
     type = R(y) * R(-x) * R(-z);
    llimit = (-45.00deg,-85.00deg,-90.00deg);
    ulimit = (45.00deg,100.00deg,94.80deg);
}
joint left_elbow {
    connect left_upper_arm.distal to left_lower_arm.proximal;
     type = R(y);
    llimit = (0.00deg);
    ulimit = (149.75deg);
    rest = (5.00deg);
    tolerance = (5.00deg);
}
joint left_wrist {
    connect left_lower_arm.distal to left_palm.base;
     type = R(y) * R(x) * R(z);
    llimit = (-45.00deg,-85.00deg,-90.00deg);
    ulimit = (45.00deg,100.00deg,94.80deg);
}
joint left_clavicle_joint {
    connect upper_torso.lclav to left_clavicle.proximal;
     type = R(x) * R(y);
    llimit = (-12.00deg,-8.00deg);
    ulimit = (25.00deg,40.00deg);
}
joint left_shoulder {
    connect left_clavicle.lateral to left_upper_arm.proximal;
     type = R(z) * R(x) * R(y);
    llimit = (-108.55deg,-48.00deg,-61.45deg);
    ulimit = (71.50deg,180.95deg,187.65deg);
}
joint linfinger02 {
    connect left_finger01.tip to left_finger02.base0;
     type = R(-x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint linfinger01 {
    connect left_finger00.tip to left_finger01.base0;
     type = R(-x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
```

```
}
joint lpinfinger31 {
    connect left_finger30.tip to left_finger31.base0;
    type = R(-x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint lpinfinger32 {
    connect left_finger31.tip to left_finger32.base0;
    type = R(-x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint lringfinger22 {
    connect left_finger21.tip to left_finger22.base0;
    type = R(-x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint lringfinger21 {
    connect left_finger20.tip to left_finger21.base0;
    type = R(-x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint lmidfinger12 {
    connect left_finger11.tip to left_finger12.base0;
    type = R(-x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint lmidfinger11 {
    connect left_finger10.tip to left_finger11.base0;
    type = R(-x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint left_finger00 {
    connect left_palm.f11 to left_finger00.base0;
    type = R(z) * R(-x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint left_finger10 {
    connect left_palm.f22 to left_finger10.base0;
    type = R(z) * R(-x);
```

```
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint left_finger20 {
    connect left_palm.f33 to left_finger20.base0;
    type = R(z) * R(-x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint left_finger30 {
    connect left_palm.f44 to left_finger30.base0;
    type = R(z) * R(-x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint lthumb2 {
    connect left_thumb1.tip to left_thumb2.base0;
    type = R(-x);
    llimit = (-75.00deg);
    ulimit = (10.00deg);
}
joint lthumb1 {
    connect left_thumb0.tip to left_thumb1.base0;
    type = R(-x);
    llimit = (-45.00deg);
    ulimit = (5.00deg);
}
joint lthumb0 {
    connect left_palm.thumb0 to left_thumb0.base0;
    type = R(-z) * R(y);
    llimit = (0.00deg,0.00deg);
    ulimit = (40.00deg,110.00deg);
}
joint right_eyeball {
    connect bottom_head.right_eyeball to right_eyeball.base;
    type = R(x) * R(z);
    llimit = (-20.00deg,-30.00deg);
    ulimit = (20.00deg,30.00deg);
}
joint left_eyeball {
    connect bottom_head.left_eyeball to left_eyeball.base;
    type = R(x) * R(z);
    llimit = (-20.00deg,-30.00deg);
    ulimit = (20.00deg,30.00deg);
}
 postureref ["mysit.post"] mysit;
```

```
    postureref ["mystand.post"] mystand;
    postureref ["functionalzero.post"] functionalzero;
    postureref ["functionalone.post"] functionalone;
    postureref ["overhead.post"] overhead;
    postureref ["sitting_forearm_up.post"] sitting_forearm_up;
    postureref ["forearm_up.post"] forearm_up;
    postureref ["forearm_up_hand_closed.post"] forearm_up_hand_closed;
    postureref ["arm_span.post"] arm_span;
    postureref ["biceps_flex.post"] biceps_flex;

    root = left_toes.distal;
}
```

APPENDIX G

LANDMARKS

LANDMARKS

This is the list of landmarks currently available with the human model. Since landmark locations are not available from anthropometric surveys, the model's landmarks have been placed by educated guesses.

Global Location of the figure's root = (0.000000,0.000002,0.000001)

segment bottom_head

Global Location of Segment's Origin = (-9.048829,152.059723,-7.979547)

landmark.bottom_head.LM_alare_l
global=(-7.377992,159.521774,0.145453) local=(8.380000,1.690000,7.170000)
landmark.bottom_head.LM_alare_r
global=(-10.757982,159.522507,0.136798) local=(8.380000,-1.690000,7.170000)
landmark.bottom_head.LM_cheilion_l
global=(-5.685760,156.569641,-0.976336) local=(7.150000,3.380000,4.260000)
landmark.bottom_head.LM_cheilion_r
global=(-12.305872,156.151367,-0.978410) local=(7.150000,-3.240000,3.840000)
landmark.bottom_head.LM_chin
global=(-8.998285,153.443954,-0.263590) local=(7.760000,0.070000,1.110000)
landmark.bottom_head.LM_crinion
global=(-9.043282,169.232651,-0.863508) local=(7.720000,0.020000,16.910000)
landmark.bottom_head.LM_ear_bottom
global=(-16.409626,159.249008,-7.062479) local=(1.190000,-7.360000,7.150000)
landmark.bottom_head.LM_ear_pt
global=(-17.775017,162.939133,-8.547649) local=(-0.160000,-8.730000,10.890000)
landmark.bottom_head.LM_ear_top
global=(-17.259325,164.584641,-6.723486) local=(1.720000,-8.210000,12.470000)
landmark.bottom_head.LM_ectocanthus
global=(-13.732141,163.456879,-1.811425) local=(6.580000,-4.670000,11.170000)
landmark.bottom_head.LM_ectoorbitale_lft
global=(-2.707700,164.235489,-3.461925) local=(4.930000,6.350000,12.010000)
landmark.bottom_head.LM_ectoorbitale_rt
global=(-15.407663,164.238266,-3.494449) local=(4.930000,-6.350000,12.010000)
landmark.bottom_head.LM_frontotemporale_l
global=(-2.879668,165.326736,-2.600483) local=(5.830000,6.180000,13.070000)
landmark.bottom_head.LM_frontotemporale_r
global=(-15.089632,165.329422,-2.631753) local=(5.830000,-6.030000,13.070000)
landmark.bottom_head.LM_glabella
global=(-9.006763,166.189636,0.235099) local=(8.710000,0.060000,13.830000)
landmark.bottom_head.LM_gonion_lft

global=(-3.259259,154.854431,-7.563533) local=(0.500000,5.790000,2.780000)
landmark.bottom_head.LM_gonion_rt
global=(-14.839226,154.856964,-7.593189) local=(0.500000,-5.790000,2.780000)
landmark.bottom_head.LM_head_top
global=(-8.953569,175.111465,-8.056120) local=(0.740000,0.090000,23.040001)
landmark.bottom_head.LM_infraorbitale_l
global=(-6.160835,161.208618,-2.502881) local=(5.790000,2.900000,8.950000)
landmark.bottom_head.LM_infraorbitale_r
global=(-11.960818,161.209900,-2.517735) local=(5.790000,-2.900000,8.950000)
landmark.bottom_head.LM_menton
global=(-8.944624,152.279022,-1.793143) local=(6.190000,0.120000,0.000000)
landmark.bottom_head.LM_otobasion_sup
global=(-15.982209,163.560471,-5.683281) local=(2.720000,-6.930000,11.410000)
landmark.bottom_head.LM_promenton
global=(-8.998311,153.444321,-0.253596) local=(7.770000,0.070000,1.110000)
landmark.bottom_head.LM_pronasale
global=(-9.012972,160.143372,2.140517) local=(10.400000,0.060000,7.720000)
landmark.bottom_head.LM_sellion
global=(-9.006626,163.246948,-0.070829) local=(8.300000,0.060000,10.900000)
landmark.bottom_head.LM_stomion
global=(-9.008897,156.262756,0.216811) local=(8.340000,0.060000,3.910000)
landmark.bottom_head.LM_subnasale
global=(-9.018757,158.529144,0.356571) local=(8.560000,0.050000,6.170000)
landmark.bottom_head.LM_top_of_head
global=(-8.910146,175.015381,-13.305922) local=(-4.510000,0.120000,23.129999)
landmark.bottom_head.LM_tragion_lft
global=(-1.600987,162.184769,-6.258206) local=(2.060000,7.450000,10.060000)
landmark.bottom_head.LM_tragion_rt
global=(-16.500944,162.188034,-6.296365) local=(2.060000,-7.450000,10.060000)
landmark.bottom_head.LM_zygion_lft
global=(-3.159506,162.213150,-2.931087) local=(5.390000,5.900000,9.970000)
landmark.bottom_head.LM_zygion_rt
global=(-14.959473,162.215744,-2.961307) local=(5.390000,-5.900000,9.970000)
landmark.bottom_head.LM_zygofrontale_lft
global=(-3.930634,166.002853,-2.166848) local=(6.290000,5.130000,13.730000)
landmark.bottom_head.LM_zygofrontale_rt
global=(-14.190604,166.005112,-2.193123) local=(6.290000,-5.130000,13.730000)

segment l3

Global Location of Segment's Origin = (-9.005613,112.955307,-8.957743)

landmark.l3.LM_tenth_rib
global=(-18.551765,113.442459,-0.858032) local=(-9.540000,0.160000,8.120000)

segment l4

Global Location of Segment's Origin = (-9.088500,109.501999,-8.871319)

landmark.l4.LM_waist_ant__naval_
global=(-9.051386,111.855400,1.663213) local=(0.000000,2.090000,10.590000)
landmark.l4.LM_waist_lft
global=(6.427049,111.221397,-8.895446) local=(15.470000,2.090000,0.000000)
landmark.l4.LM_waist_post
global=(-9.086878,111.353027,-18.490587) local=(-0.060000,2.090000,-9.570000)
landmark.l4.LM_waist_rt
global=(-24.504089,111.960106,-8.951501) local=(-15.470000,2.090000,0.000000)

segment l5

Global Location of Segment's Origin = (-9.205016,105.851471,-8.522093)

landmark.l5.LM_iliocristale
global=(-23.001032,106.772087,-5.095790) local=(-13.810000,0.400000,3.470000)
landmark.l5.LM_waist_ant
global=(-9.096566,109.940613,1.359975) local=(0.010000,3.740000,10.020000)
landmark.l5.LM_waist_lft
global=(6.147124,109.134293,-8.993267) local=(15.250000,3.740000,-0.370000)
landmark.l5.LM_waist_rt
global=(-24.330116,110.014496,-9.053874) local=(-15.240000,3.740000,-0.370000)

segment left_clavicle

Global Location of Segment's Origin = (-8.179223,141.038269,-12.673971)

landmark.left_clavicle.LM_clavicle_pt_l
global=(10.592699,147.613678,-8.927941) local=(2.990000,-6.930000,18.780001)
landmark.left_clavicle.LM_clavicle_pt_lft
global=(8.543008,149.079590,-12.831386) local=(-1.040000,-7.980000,16.719999)
landmark.left_clavicle.LM_midscye_lft
global=(8.284431,141.542007,-17.937860) local=(-5.330000,0.050000,16.450001)
landmark.left_clavicle.LM_post_scye_lft
global=(2.668499,134.764145,-16.211246) local=(-2.890000,6.610000,10.840000)

segment left_finger12

Global Location of Segment's Origin = (11.509692,71.208534,-15.926214)

landmark.left_finger12.LM_dactylion_III_1
global=(12.423116,67.881714,-15.875632) local=(0.010000,3.370000,0.740000)

segment left_foot

Global Location of Segment's Origin = (3.311255,4.943436,-20.414867)

landmark.left_foot.LM_heel_pt_lft_med
global=(-1.829579,0.039345,-18.879827) local=(1.560000,5.030000,5.010000)

segment left_upper_arm

Global Location of Segment's Origin = (10.451850,147.819794,-12.718535)

landmark.left_upper_arm.LM_acromion_1
global=(10.968409,147.874466,-13.238456) local=(-0.524112,-0.515213,0.000000)
landmark.left_upper_arm.LM_deltoid_pt_lft
global=(13.061237,138.808868,-8.492590) local=(5.137892,-2.622178,8.520000)

segment lower_torso

Global Location of Segment's Origin = (-10.055210,80.758568,-7.939765)

landmark.lower_torso.LM_ant_sup_iliac_spine_lft
global=(2.064229,96.984665,1.969514) local=(10.590000,11.580000,16.190001)
landmark.lower_torso.LM_ant_sup_iliac_spine_rt
global=(-21.202429,97.797142,2.317561) local=(10.990000,-11.700000,16.190001)
landmark.lower_torso.LM_buttock_pt_lft_lat
global=(5.738100,91.488274,-11.333358) local=(-2.940000,15.410000,11.400000)
landmark.lower_torso.LM_buttock_pt_post
global=(-9.578417,93.918655,-20.107567) local=(-11.590000,0.000000,13.680000)
landmark.lower_torso.LM_buttock_pt_rt_lat
global=(-25.063843,92.541100,-11.401647) local=(-2.940000,-15.410000,11.400000)
landmark.lower_torso.LM_gluteal_furrow_pt2

global=(-18.908493,83.789276,-16.234585) local=(-8.150000,-8.970000,3.080000)
landmark.lower_torso.LM_post_iliac_spine
global=(-12.733405,102.325615,-19.961895) local=(-11.080000,-3.440000,21.959999)
landmark.lower_torso.LM_trochanter
global=(-26.013121,91.313606,-8.336633) local=(0.070000,-16.309999,10.010000)
landmark.lower_torso.LM_trochanterion
global=(-26.732830,88.861969,-5.839246) local=(2.460000,-16.940001,7.430000)

segment neck

Global Location of Segment's Origin = (-8.715386,143.008820,-10.099344)

landmark.neck.LM_Infrathyroid
global=(-8.728626,150.549652,-4.282890) local=(6.080000,0.000000,7.330000)
landmark.neck.LM_cervicale
global=(-8.636642,154.521683,-16.431067) local=(-5.920000,0.060000,11.730000)
landmark.neck.LM_cervicale0
global=(-8.706629,146.722916,-17.105347) local=(-6.870000,-0.010000,3.960000)
landmark.neck.LM_neck_lft_lat
global=(-3.727130,147.737350,-5.101000) local=(5.150000,5.000000,4.550000)
landmark.neck.LM_neck_rt_lat
global=(-13.727101,147.739548,-5.126610) local=(5.150000,-5.000000,4.550000)
landmark.neck.LM_submandibular
global=(-8.983592,152.316559,-6.097291) local=(4.330000,-0.260000,9.160000)
landmark.neck.LM_trapezius_pt_lft
global=(-3.387363,152.310547,-12.426798) local=(-2.010000,5.320000,9.380000)
landmark.neck.LM_trapezius_pt_rt
global=(-14.027332,152.312897,-12.454046) local=(-2.010000,-5.320000,9.380000)

segment right_clavicle

Global Location of Segment's Origin = (-9.319221,141.038498,-12.676890)

landmark.right_clavicle.LM_clavicle_pt_r
global=(-28.137142,147.624252,-9.047018) local=(2.970000,6.930000,18.809999)
landmark.right_clavicle.LM_clavicle_pt_rt
global=(-26.036970,149.084030,-12.890108) local=(-1.010000,7.980000,16.719999)
landmark.right_clavicle.LM_midscye_rt
global=(-25.755480,141.648926,-18.014584) local=(-5.330000,0.050000,16.450001)
landmark.right_clavicle.LM_midshoulder
global=(-20.427019,150.267334,-12.761308) local=(-1.020000,9.170000,11.110000)
landmark.right_clavicle.LM_post_scye_rt

global=(-20.151440,134.769135,-16.269688) local=(-2.890000,-6.610000,10.840000)

segment right_finger00

Global Location of Segment's Origin = (-28.364407,79.541641,-14.074752)

landmark.right_finger00.LM_metacarpale_II
global=(-28.485756,78.326057,-12.563277) local=(1.496800,1.225800,-0.184178)

segment right_finger02

Global Location of Segment's Origin = (-28.316502,72.356384,-13.326163)

landmark.right_finger02.LM_dactylion_II
global=(-28.969954,68.965134,-13.445720) local=(-0.160000,3.351200,-0.828100)

segment right_finger12

Global Location of Segment's Origin = (-28.261786,71.207993,-16.027599)

landmark.right_finger12.LM_dactylion_III_r
global=(-28.815868,67.794975,-15.986826) local=(0.000000,3.380000,-0.730000)

segment right_finger30

Global Location of Segment's Origin = (-28.139170,80.313751,-20.830462)

landmark.right_finger30.LM_metacarpale_V
global=(-27.982082,79.315765,-21.752008) local=(-0.933584,0.993600,0.105264)

segment right_foot

Global Location of Segment's Origin = (-19.539377,5.287267,-20.619473)

landmark.right_foot.LM_fifth_metatarsophalangeal_protrusion
global=(-24.962395,2.223139,-1.533533) local=(19.100000,5.030000,3.600000)

landmark.right_foot.LM_first_metatarsophalangeal_protrusion
global=(-15.127543,3.446063,-1.077419) local=(19.450001,-4.750000,1.970000)
landmark.right_foot.LM_heel_pt_rt_lat
global=(-19.683920,0.200716,-19.845764) local=(0.690000,-0.080000,5.100000)
landmark.right_foot.LM_heel_pt_rt_med
global=(-14.749946,0.007158,-18.908821) local=(1.560000,-5.030000,5.100000)
landmark.right_foot.LM_pternion
global=(-16.325338,2.720142,-20.535004) local=(0.000000,-3.320000,2.430000)

segment right_lower_arm

Global Location of Segment's Origin = (-28.102646,117.453552,-17.217857)

landmark.right_lower_arm.LM_elbow_crease
global=(-28.115301,109.562454,-12.953102) local=(4.401768,0.000000,7.815500)
landmark.right_lower_arm.LM_stylion
global=(-27.548088,86.411102,-14.631917) local=(3.125594,-0.568000,30.992500)

segment right_lower_leg

Global Location of Segment's Origin = (-18.138632,56.291206,-5.633486)

landmark.right_lower_leg.LM_calf
global=(-23.894167,39.621296,-10.050694) local=(-0.790000,6.010000,17.139999)
landmark.right_lower_leg.LM_dorsal_juncture
global=(-18.138632,56.291206,-5.633486) local=(0.000000,0.000000,0.000000)
landmark.right_lower_leg.LM_knee_pt_ant
global=(-18.236721,50.994797,-0.524657) local=(6.110000,0.140000,4.100000)
landmark.right_lower_leg.LM_lateral_malleolus
global=(-19.608755,6.714833,-15.664397) local=(0.640000,2.210000,50.549999)
landmark.right_lower_leg.LM_medial_malleolus
global=(-15.478425,6.726800,-15.669661) local=(0.620000,-1.920000,50.599998)
landmark.right_lower_leg.LM_midpatella
global=(-17.982853,48.559208,-0.598526) local=(6.550000,-0.080000,6.500000)
landmark.right_lower_leg.LM_suprapatella
global=(-18.211292,49.708824,-0.157180) local=(6.740000,0.130000,5.280000)

segment right_thumb2

Global Location of Segment's Origin = (-27.755793,81.935722,-7.147518)

landmark.right_thumb2.LM_thumbtip_
global=(-28.475712,80.567024,-2.897352) local=(-0.170000,4.450000,0.790000)

segment right_toes

Global Location of Segment's Origin = (-20.143847,2.100529,-1.459460)

landmark.right_toes.LM_acropodion
global=(-20.540081,2.426763,5.881610) local=(7.350000,0.310000,-0.190000)

segment right_upper_arm

Global Location of Segment's Origin = (-27.946840,147.828217,-12.816874)

landmark.right_upper_arm.LM_acromion_r
global=(-28.460705,147.883102,-13.339434) local=(-0.524112,0.515213,0.000000)
landmark.right_upper_arm.LM_ant_scye_upper_arm
global=(-25.785700,136.179886,-9.580854) local=(4.430202,-2.171978,11.246401)
landmark.right_upper_arm.LM_biceps_pt
global=(-31.513697,125.747787,-8.139332) local=(6.968899,3.550011,21.470402)
landmark.right_upper_arm.LM_deltoid_pt_rt
global=(-30.581797,138.818451,-8.604359) local=(5.137892,2.622178,8.520000)
landmark.right_upper_arm.LM_radiale
global=(-33.775520,113.961807,-16.638086) local=(-0.245600,5.831000,34.080002)

segment right_upper_leg

Global Location of Segment's Origin = (-17.989824,93.975716,-7.221995)

landmark.right_upper_leg.LM_dorsal_junct
global=(-18.320169,61.901680,-17.174091) local=(-9.720000,1.040000,32.130001)
landmark.right_upper_leg.LM_dorsal_juncture
global=(-18.146263,56.930321,-13.788193) local=(-6.300000,0.940000,37.080002)
landmark.right_upper_leg.LM_lat_femoral_epicondyle__standing
global=(-22.336010,50.980331,-8.750582) local=(-1.190000,5.210000,42.910000)

segment t1

Global Location of Segment's Origin = (-8.760646,147.299805,-7.678009)

landmark.t1.LM_neck_ant
global=(-8.732594,147.216461,-3.020071) local=(0.040000,1.930000,4.240000)
landmark.t1.LM_suprasternale
global=(-8.771486,147.499634,-3.428000) local=(0.000000,2.010000,3.750000)

segment t8

Global Location of Segment's Origin = (-8.784616,133.208450,-8.145068)

landmark.t8.LM_ant_scye_torso
global=(-25.314480,135.896225,-7.724455) local=(-16.540001,2.640000,0.310000)
landmark.t8.LM_midspine
global=(-8.760368,135.234741,-19.636961) local=(0.000000,1.440000,-11.580000)
landmark.t8.LM_midspine0
global=(-8.766079,133.805603,-19.689629) local=(0.000000,0.010000,-11.560000)

Number of segments: 24
Number of LM sites: 106

NO. OF COPIES	ORGANIZATION
2	ADMINISTRATOR DEFENSE TECHNICAL INFO CENTER ATTN DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	DIRECTOR US ARMY RESEARCH LABORATORY ATTN AMSRL CS AL TA RECORDS MANAGEMENT 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LABORATORY ATTN AMSRL CI LL TECHNICAL LIBRARY 2800 POWDER MILL RD ADELPHI MD 207830-1197
1	DIRECTOR US ARMY RESEARCH LABORATORY ATTN AMSRL CS AL TP TECH PUBLISHING BRANCH 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTORATE FOR MANPRINT ATTN DAPE MR DEPUTY CHIEF OF STAFF PERSONNEL 300 ARMY PENTAGON WASHINGTON DC 20310-0300
1	OUSD(A)/DDDR&E(R&A)/E&LS PENTAGON ROOM 3D129 WASHINGTON DC 20301-3080
1	CODE 1142PS OFFICE OF NAVAL RESEARCH 800 N QUINCY STREET ARLINGTON VA 22217-5000
1	DR ARTHUR RUBIN NATL INST OF STANDARDS & TECH BUILDING 226 ROOM A313 GAITHERSBURG MD 20899
1	COMMANDER US ARMY RESEARCH INSTITUTE ATTN PERI ZT (DR E M JOHNSON) 5001 EISENHOWER AVENUE ALEXANDRIA VA 22333-5600

NO. OF COPIES	ORGANIZATION
1	DEFENSE LOGISTICS STUDIES INFORMATION EXCHANGE ATTN DIRECTOR DLSIE ATSZ DL BLDG 12500 2401 QUARTERS ROAD FORT LEE VA 23801-1705
1	DEPUTY COMMANDING GENERAL ATTN EXS (Q) MARINE CORPS RD&A COMMAND QUANTICO VA 22134
1	HEADQUARTERS USATRADOC ATTN ATCD SP FORT MONROE VA 23651
1	COMMANDER USATRADOC COMMAND SAFETY OFFICE ATTN ATOS (MR PESSAGNO/MR LYNE) FORT MONROE VA 23651-5000
1	COMMANDER US ARMY MATERIEL COMMAND ATTN AMCAM 5001 EISENHOWER AVENUE ALEXANDRIA VA 22333-0001
1	DIRECTOR TDAD DCST ATTN ATTG C BLDG 161 FORT MONROE VA 23651-5000
1	COMMANDER USA OPERATIONAL TEST & EVAL AGENCY ATTN CSTE TSM 4501 FORD AVE ALEXANDRIA VA 22302-1458
1	USA BIOMEDICAL R&D LABORATORY ATTN LIBRARY FORT DETRICK BUILDING 568 FREDERICK MD 21702-5010
1	HQ USAMRDC ATTN SGRD PLC FORT DETRICK MD 21701
1	COMMANDER USA AEROMEDICAL RESEARCH LAB ATTN LIBRARY FORT RUCKER AL 36362-5292

NO. OF COPIES	ORGANIZATION
1	US ARMY SAFETY CENTER ATTN CSSC SE FORT RUCKER AL 36362
1	CHIEF ARMY RESEARCH INSTITUTE AVIATION R&D ACTIVITY ATTN PERI IR FORT RUCKER AL 36362-5354
1	AIR FORCE FLIGHT DYNAMICS LAB ATTN AFWAL/FIES/SURVIAC WRIGHT PATTERSON AFB OH 45433
1	AAMRL/HE WRIGHT PATTERSON AFB OH 45433-6573
1	US ARMY NATICK RD&E CENTER ATTN STRNC YBA NATICK MA 01760-5020
1	US ARMY TROOP SUPPORT CMD NATICK RD&E CENTER ATTN BEHAVIORAL SCI DIV SSD NATICK MA 01760-5020
1	US ARMY TROOP SUPPORT CMD NATICK RD&E CENTER ATTN TECH LIBRARY (STRNC MIL) NATICK MA 01760-5040
1	DR RICHARD JOHNSON HEALTH & PERFORMANCE DIVISION US ARIEM NATICK MA 01760-5007
1	LOCKHEED SANDERS INC BOX MER 24 1583 NASHUA NH 03061-0868
1	MEDICAL LIBRARY BLDG 148 NAVAL SUBMARINE MEDICAL RSCH LAB BOX 900 SUBMARINE BASE NEW LONDON GROTON CT 06340
1	DR JON FALLESEN ARI FIELD UNIT PO BOX 3407 FORT LEAVENWORTH KS 66027-0347
1	COMMANDER USAMC LOGISTICS SUPPORT ACTIVITY ATTN AMXLS AE REDSTONE ARSENAL AL 35898-7466

NO. OF COPIES	ORGANIZATION
1	ARI FIELD UNIT FORT KNOX BUILDING 2423 PERI IK FORT KNOX KY 40121-5620
1	COMMANDANT USA ARTILLERY & MISSILE SCHOOL ATTN USAAMS TECH LIBRARY FORT SILL OK 73503
1	COMMANDER WHITE SANDS MISSILE RANGE ATTN STEWS TE RE WHITE SANDS MISSILE RANGE NM 88002
1	COMMANDER WHITE SANDS MISSILE RANGE ATTN TECHNICAL LIBRARY WHITE SANDS MISSILE RANGE NM 88002
1	USA TRADOC ANALYSIS COMMAND ATTN ATRC WSR (D ANGUIANO) WHITE SANDS MISSILE RANGE NM 88002-5502
1	STRICOM 12350 RESEARCH PARKWAY ORLANDO FL 32826-3276
1	COMMANDER USA TANK-AUTOMOTIVE R&D CENTER ATTN AMSTA TSL (TECH LIBRARY) WARREN MI 48397-5000
1	COMMANDER USA COLD REGIONS TEST CENTER ATTN STECR TS A APO AP 96508-7850
1	US ARMY RSCH DEV STDZN GP-UK ATTN DR MIKE STOUT PSC 802 BOX 15 FPO AE 09499-1500
1	INSTITUTE FOR DEFENSE ANALYSES ATTN DR JESSE ORLANSKY 1801 N BEAUREGARD STREET ALEXANDRIA VA 22311
1	PURDUE UNIVERSITY SERIALS UNIT CDM KARDEX 1535 STEWART CENTER WEST LAFAYETTE IN 47907-1535

NO. OF COPIES	ORGANIZATION
1	GOVT PUBLICATIONS LIBRARY 409 WILSON M UNIVERSITY OF MINNESOTA MINNEAPOLIS MN 55455
1	DR HARVEY A TAUB RSCH SECTION PSYCH SECTION VETERANS ADMIN HOSPITAL IRVING AVENUE & UNIVERSITY PLACE SYRACUSE NY 13210
1	DR ANTHONY DEBONS IDIS UNIVERSITY OF PITTSBURGH PITTSBURGH PA 15260
1	MR R BEGGS BOEING-HELICOPTER CO P30-18 PO BOX 16858 PHILADELPHIA PA 19142
1	MR LARRY W AVERY BATTELLE PACIFIC NW LABS PO BOX 999 MAIL STOP K6-66 RICHLAND WA 99352
1	LIBRARY ESSEX CORPORATION SUITE 510 1430 SPRING HILL ROAD MCLEAN VA 22102-3000
1	DR BEN B MORGAN DEPARTMENT OF PSYCHOLOGY UNIVERSITY OF CENTRAL FLORIDA PO BOX 25000 ORLANDO FL 32816
1	DR ARTHUR S KAMLET BELL LABORATORIES 6200 EAST BROAD STREET COLUMBUS OH 43213
1	GENERAL MOTORS CORPORATION NORTH AMERICAN OPERATIONS PORTFOLIO ENGINEERING CENTER HUMAN FACTORS ENGINEERING ATTN MR A J ARNOLD STAFF PROJ ENG ENGINEERING BLDG 30200 MOUND RD BOX 9010 WARREN MI 48090-9010
1	GENERAL DYNAMICS LAND SYSTEMS DIV LIBRARY PO BOX 1901 WARREN MI 48090
1	DR PAUL R MCCRIGHT INDUSTRIAL ENGINEERING DEPT KANSAS STATE UNIVERSITY MANHATTA KS 66502
1	DR MM AYOUB DIRECTOR INST FOR ERGONOMICS RESEARCH TEXAS TECH UNIVERSITY LUBBOCK TX 79409
1	MR KENNETH C CROMBIE TECHNICAL LIBRARIAN E104 DELCO SYSTEMS OPERATIONS 6767 HOLLISTER AVENUE GOLETA CA 93117
1	MR WALT TRUSZKOWSKI CODE 522.3 NASA/GODDARD SPACE FLIGHT CENTER GREENBELT MD 20771
1	DIRECTOR US ARMY AEROFLIGHT DYNAMICS DIR ATTN SAVRT AF D (A W KERR) AMES RESEARCH CENTER (MS 215-1) MOFFETT FIELD CA 94035-1099
1	DR NORMAN BADLER DEPT OF COMPUTER & INFORMATION SCIENCE UNIVERSITY OF PENNSYLVANIA PHILADELPHIA PA 19104-6389
1	COMMANDER US ARMY RESEARCH INSTITUTE OF ENVIRONMENTAL MEDICINE NATICK MA 01760-5007
1	DR DANIEL J POND BATTELLE PNL/K6-66 PO BOX 999 RICHLAND WA 99350
1	HQDA (DAPE ZXO) ATTN DR FISCHL WASHINGTON DC 20310-0300
1	HUMAN FACTORS ENG PROGRAM DEPT OF BIOMEDICAL ENGINEERING COLLEGE OF ENGINEERING & COMPUTER SCIENCE WRIGHT STATE UNIVERSITY DAYTON OH 45435

NO. OF COPIES	ORGANIZATION
1	COMMANDER USA MEDICAL R&D COMMAND ATTN SGRD PLC (LTC K FRIEDL) FORT DETRICK MD 21701-5012
1	PEO STANDARD ARMY MGMT INFORMATION SYSTEM ATTN AS PES STOP C-3 FT BELVOIR VA 22060-5456
1	PEO ARMORED SYS MODERNIZATION US ARMY TANK-AUTOMOTIVE CMD ATTN SFAE ASM S WARREN MI 48397-5000
1	PEO COMBAT SUPPORT ATTN AMCPEO CS US ARMY TANK AUTOMOTIVE CMD WARREN MI 48397-5000
1	PEO MGMT INFORMATION SYSTEMS ATTN AS PEM STOP C-2 BLDG 1465 FT BELVOIR VA 22060-5456
1	PEO ARMAMENTS ATTN AMCPEO AR BLDG 171 PICATINNY ARSENAL NJ 07806-5000
1	PEO INTELLIGENCE & ELECTRONIC WARFARE ATTN AMCPEO IEW VINT HILL FARMS STATION BLDG 197 WARRENTON VA 22186-5115
1	PEO COMMUNICATIONS ATTN SFAE CM RE FT MONMOUTH NJ 07703-5000
1	PEO AIR DEFENSE ATTN SFAE AD S US ARMY MISSILE COMMAND REDSTONE ARSENAL AL 35898-5750
1	PEO STRATEGIC DEFENSE PO BOX 15280 ATTN DASD ZA US ARMY STRATEGIC DEFENSE CMD ARLINGTON VA 22215-0280
1	PROGRAM MANAGER RAH-66 4300 GOODFELLOW BLVD ST LOUIS MO 63120-1798

NO. OF COPIES	ORGANIZATION
1	DENNIS L SCHMICKLY CREW SYSTEMS ENGINEERING MCDONNELL DOUGLAS HELICOPTER 5000 EAST MCDOWELL ROAD MESA AZ 85205-9797
1	JON TATRO HUMAN FACTORS SYSTEM DESIGN BELL HELICOPTER TEXTRON INC PO BOX 482 MAIL STOP 6 FT WORTH TX 76101
1	CHIEF CREW SYSTEMS INTEGRATION SIKORSKY AIRCRAFT M/S S3258 NORTH MAIN STREET STRATFORD CT 06602
1	GENERAL ELECTRIC COMPANY ARMAMENT SYSTEMS DEPT RM 1309 ATTN HF/MANPRINT R C MCLANE LAKESIDE AVENUE BURLINGTON VT 05401-4985
1	OASD (FM&P) WASHINGTON DC 20301-4000
1	COMMANDER US ARMY MATERIEL COMMAND ATTN AMCDE AQ 5001 EISENHOWER AVENUE ALEXANDRIA VA 22333
1	COMMANDANT US ARMY ARMOR SCHOOL ATTN ATSB CDS (MR LIPSCOMB) FT KNOX KY 40121-5215
1	COMMANDER US ARMY AVIATION CENTER ATTN ATZQ CDM S (MR MCCRACKEN) FT RUCKER AL 36362-5163
1	COMMANDER US ARMY SIGNAL CTR & FT GORDON ATTN ATZH CDM FT GORDON GA 30905-5090
1	DIRECTOR US ARMY AEROFLIGHT DYNAMICS DIR MAIL STOP 239-9 NASA AMES RESEARCH CENTER MOFFETT FIELD CA 94035-1000

NO. OF COPIES	ORGANIZATION
1	PROJECT MANAGER SIGNALS WARFARE ATTN SFAE IEW SG (ALAN LINDLEY) BLDG P-181 VINT HILL FARMS STATION WARRENTON VA 22186-5116
1	COMMANDER MARINE CORPS SYSTEMS COMMAND ATTN CBGT QUANTICO VA 22134-5080
1	DIRECTOR AMC-FIELD ASSIST IN SCIENCE & TECHNOLOGY ATTN AMC-FAST (RICHARD FRANSEEN) FT BELVOIR VA 22060-5606
1	COMMANDER US ARMY FORCES COMMAND ATTN FCDJ SA BLDG 600 AMC FAST SCIENCE ADVISER FT MCPHERSON GA 30330-6000
1	COMMANDER I CORPS AND FORT LEWIS AMC FAST SCIENCE ADVISER ATTN AFZH CSS FORT LEWIS WA 98433-5000
1	HQ III CORPS & FORT HOOD OFFICE OF THE SCIENCE ADVISER ATTN AFZF CS SA FORT HOOD TX 76544-5056
1	COMMANDER U.S. ARMY NATL TRAINING CENTER AMC FAST SCIENCE ADVISER ATTN AMXLA SA FORT IRWIN CA 92310
1	COMMANDER HQ XVIII ABN CORPS & FORT BRAGG OFFICE OF THE SCI ADV BLDG 1-1621 ATTN AFZA GD FAST FORT BRAGG NC 28307-5000
1	SOUTHCOM WASHINGTON FIELD OFC 1919 SOUTH EADS ST SUITE L09 AMC FAST SCIENCE ADVISER ARLINGTON VA 22202
1	HQ US SPECIAL OPERATIONS CMD AMC FAST SCIENCE ADVISER ATTN SOSD MACDILL AIR FORCE BASE TAMPA FL 33608-0442

NO. OF COPIES	ORGANIZATION
1	HQ US ARMY EUROPE AND 7TH ARMY ATTN AEAGX SA OFFICE OF THE SCIENCE ADVISER APO AE 09014
1	COMMANDER HQ 21ST THEATER ARMY AREA CMD AMC FAST SCIENCE ADVISER ATTN AERSA APO AE 09263
1	COMMANDER HEADQUARTERS USEUCOM AMC FAST SCIENCE ADVISER UNIT 30400 BOX 138 APO AE 09128
1	HQ 7TH ARMY TRAINING COMMAND UNIT #28130 AMC FAST SCIENCE ADVISER ATTN AETT SA APO AE 09114
1	COMMANDER HHC SOUTHERN EUROPEAN TASK FORCE ATTN AESE SA BUILDING 98 AMC FAST SCIENCE ADVISER APO AE 09630
1	COMMANDER US ARMY PACIFIC AMC FAST SCIENCE ADVISER ATTN APSA FT SHAFTER HI 96858-5L00
1	COMMANDER US ARMY JAPAN/IX CORPS UNIT 45005 ATTN APAJ SA AMC FAST SCIENCE ADVISERS APO AP 96343-0054
1	AMC FAST SCIENCE ADVISERS PCS #303 BOX 45 CS-SO APO AP 96204-0045
1	COMMANDER ALASKAN COMMAND ATTN SCIENCE ADVISOR (MR GRILLS) 6-900 9TH ST STE 110 ELMENDORF AFB ALASKA 99506
1	CDR & DIR USAE WATERWAYS EXPERIMENTAL STATION CD DEPT #1153 ATTN CEWES IM MI R (A S CLARK) 3909 HALLS FERRY ROAD VICKSBURG MS 39180-6199

NO. OF COPIES	ORGANIZATION
1	MR GARY PERLMAN COMPUTER & INFORMATION SCIENCE OHIO STATE UNIV RM 228 BOLZ HALL 2036 NEIL AVENUE COLUMBUS OH 43210-1277
1	DR SEHCHANG HAH DEPT OF BEHAVIORAL SCIENCES & LEADERSHIP BUILDING 601 ROOM 281 US MILITARY ACADEMY WEST POINT NEW YORK 10996-1784
1	US ARMY RESEARCH INSTITUTE ATTN PERI IK (DOROTHY L FINLEY) 2423 MORANDE STREET FORT KNOX KY 40121-5620
1	DENNIS SCHMIDT HQDA DAMO FDQ 400 ARMY PENTAGON WASHINGTON DC 20310-0460
1	US MILITARY ACADEMY MATHEMATICAL SCIENCES CENTER OF EXCELLENCE DEPT OF MATHEMATICAL SCIENCES ATTN MDN A MAJ DON ENGEN THAYER HALL WEST POINT NY 10996-1786
1	ARL HRED USAADASCH FLD ELEMENT ATTN AMSRL HR ME (K REYNOLDS) ATTN ATSA CD 5800 CARTER ROAD FORT BLISS TX 79916-3802
1	ARL HRED ARDEC FIELD ELEMENT ATTN AMSRL HR MG (R SPINE) BUILDING 333 PICATINNY ARSENAL NJ 07806-5000
1	ARL HRED ARMC FIELD ELEMENT ATTN AMSRL HR MH (J JOHNSON) BLDG 1109B 3RD FLOOR FT KNOX KY 40121-5215
1	ARL HRED AVNC FIELD ELEMENT ATTN AMSRL HR MJ (R ARMSTRONG) PO BOX 620716 BUILDING 514 FT RUCKER AL 36362-0716

NO. OF COPIES	ORGANIZATION
1	ARL HRED FT BELVOIR FIELD ELEMENT ATTN AMSRL HR MK (P SCHOOL) 10109 GRIDLEY ROAD SUITE A102 FORT BELVOIR VA 22060-5850
1	ARL HRED CECOM FIELD ELEMENT ATTN AMSRL HR ML (J MARTIN) MYER CENTER RM 3C214 FT MONMOUTH NJ 07703-5630
1	ARL HRED FT HOOD FIELD ELEMENT ATTN AMSRL HR MV (E SMOOTZ) HQ TEXCOM BLDG 91012 RM 111 FT HOOD TX 76544-5065
1	ARL HRED MICOM FIELD ELEMENT ATTN AMSRL HR MO (T COOK) BUILDING 5400 ROOM C242 REDSTONE ARSENAL AL 35898-7290
2	ARL HRED NATICK FIELD ELEMENT ATTN AMSRL HR MQ (M FLETCHER) ATTN SSCNC A (D SEARS) USASSCOM NRDEC BLDG 3 RM R-140 NATICK MA 01760-5015
1	ARL HRED SC&FG FIELD ELEMENT ATTN AMSRL HR MS (L BUCKALEW) SIGNAL TOWERS RM 207 FORT GORDON GA 30905-5233
1	ARL HRED TACOM FIELD ELEMENT ATTN AMSRL HR MU (M SINGAPORE) BLDG 200A 2ND FLOOR WARREN MI 48397-5000
1	ARL HRED STRICOM FIELD ELEMENT ATTN AMSRL HR MT (A GALBAVY) 12350 RESEARCH PARKWAY ORLANDO FL 32826-3276
1	ARL HRED USAIC FIELD ELEMENT ATTN AMSRL HR MW (E REDDEN) BLDG 4 ROOM 332 FT BENNING GA 31905-5400
1	ARL HRED USAFAS FIELD ELEMENT ATTN AMSRL HR MF (L PIERCE) BLDG 3040 RM 220 FORT SILL OK 73503-5600
1	ARL HRED USASOC FIELD ELEMENT ATTN AMSRL HR MN (F MALKIN) HQ USASOC BLDG E2929 FORT BRAGG NC 28307-5000

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1997	Final

4. TITLE AND SUBTITLE

Efforts in Preparation for Jack Validation

5. FUNDING NUMBERS

AMS: 622716.H700011
PR: 1L162716AH70
PE: 6.27.16
Contract No. DAMD 17 94 J 4486

6. AUTHOR(S)

Azuola, F.E.; Badler, N.L.; Ho, P.-H.; Huh, S.; Kokkevis, E.; Ting, B.-J.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, Pennsylvania 19104-6389

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
Human Research & Engineering Directorate
Aberdeen Proving Ground, MD 21005-5425

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARL-CR-418

11. SUPPLEMENTARY NOTES

The contracting officer's representative (COR) is John Lockett, U.S. Army Research Laboratory, ATTN: AMSRL-HR-MB, Aberdeen Proving Ground, MD 21005-5425 (telephone 410-278-5875).

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document presents a detailed record of the methodologies, assumptions, limitations, and references used in creating the human figure model in Jack, a program that displays and manipulates articulated geometric figures. This report reflects current efforts to develop and refine Jack software to enable its validation and verification as a tool for performing human engineering analysis. These efforts include human figure model improvements, statistical anthropometric data processing methods, enhanced human figure model construction and measuring methods, and automated accomodation analysis.

This report discusses basic details of building human models, model anthropometry, scaling, Jack anthropometry-based human models, statistical data processing, figure generation tools, anthropometric errors, inverse dynamics, smooth skin implementation, guidelines used in estimating landmark locations on the model, and recommendations for validating and verifying the Jack human figure model.

14. SUBJECT TERMS

anthropometry
ergonomic validation
human figure modeling

15. NUMBER OF PAGES
165

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	