1-1-1985

TEMPUS: A System for the Design and Simulation of Human Figures in a Task-Oriented Environment

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

Jonathan Korein
University of Pennsylvania

James U. Korein
IBM Research Center

Gerald M. Radack
Case Western Reserve University

Lynne Shapiro Brotman
AT&T Bell Laboratories

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-85-20.

TEMPUS: A System for the Design and Simulation of Human Figures in a Task-Oriented Environment

Abstract

A system called TEMPUS is outlined which is being developed to simulate graphically the task-oriented activities of several human agents in a three-dimensional environment. TEMPUS is a task simulation facility for the evaluation of complex workstations vis-a-vis the normal and emergency procedures they are intended to support and the types and number of individuals who must carry them out. TEMPUS allows a user to interactively:

- Create on or more human figures which are correctly scaled according to a specific population, or which meet certain size constraints.
- View the human figure in any of several graphical modes: stick figure, line or shaded polygons, or shaded BUBBLEPERSON.
- Position the figure in any admissible position within joint angle constraints, and with the assistance of a robotics reach positioning algorithm for limbs.
- Combine the figures with three-dimensional polyhedral objects derived from an existing CAD system.
- Create shaded graphics images of bodies in such environments.
- Use all TEMPUS features in an extensible and uniform user-friendly interactive system which does not require any explicitly programming knowledge.

Other features of TEMPUS and differences between TEMPUS and other available body modeling systems are also discussed.

Keywords

Human figure modeling, interactive systems, 3D graphics, robotics, applications

Disciplines

Computer Engineering | Computer Sciences

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-85-20.

TEMPUS: A System for the Design and Simulation of Human Figures in a Task-Oriented Environment

N. I. Badler, J. Korein,
J. U. Korein, G. Radack, L. Shapiro Brotman

MS-CIS-85-20

Department of Computer and Information Science
Moore School/D2
University of Pennsylvania
Philadelphia, PA 19104

1985

* Acknowledgement: This research was supported in part by NSF grants MCS8219196-CER, MCS-82-07294, 1 RO1-HL-29985-01, U.S. Army grants DAA6-29-84-K-0061, U.S. Air Force grant 82-NM-299, AI Center grants NSF-MCS-83-05221, US Army Research office grant ARO-DAA29-84-9-0027, Lord Corporation, RCA, and Digital Equipment Corporation.

MS-C15-85-20

0-87664-872-3/85/251-257/$0 + .50PP

TEMPUS:
A SYSTEM FOR THE DESIGN AND SIMULATION OF HUMAN FIGURES IN A TASK-ORIENTED ENVIRONMENT

Norman I. Badler*
Jonathan Korein*
James U. Korein**
Gerald M. Radack***
Lynne Shapiro Brotman****

*Computer and Information Science, Moore School D2
University of Pennsylvania, Philadelphia, PA 19104

**IBM Research Center, Yorktown Heights, NY 10598

***Computer Engineering and Science
Case Western Reserve University, Cleveland, OH 44106

****AT&T Bell Laboratories, Murray Hill, NJ 07974

ABSTRACT

A system called TEMPUS is outlined which is being developed to simulate graphically the task-oriented activities of several human agents in a three-dimensional environment. TEMPUS is a task simulation facility for the evaluation of complex workstations vis-a-vis the normal and emergency procedures they are intended to support and the types and number of individuals who must carry them out. TEMPUS allows a user to interactively:

* Create one or more human figures which are correctly scaled according to a specific population, or which meet certain size constraints.
* View the human figure in any of several graphical modes: stick figure, line or shaded polygons, or shaded BUBBLEPERSON.
* Position the figure in any admissible position within joint angle constraints, and with the assistance of a robotics reach positioning algorithm for limbs.
* Combine the figures with three-dimensional polyhedral objects derived from an existing CAD system.
* Create shaded graphics images of bodies in such environments.
* Use all TEMPUS features in an extensible and uniform user-friendly interactive system which does not require any explicit programming knowledge.

Other features of TEMPUS and differences between TEMPUS and other available body modeling systems are also discussed.

Key words and phrases: Human figure modeling, interactive systems, 3D graphics, robotics, applications.

INTRODUCTION

TEMPUS is an interactive system under development at the University of Pennsylvania for the graphical simulation of the activities of several human agents in a workstation environment. TEMPUS is a task simulation facility (supported by the Crew Station Design Section of the NASA Johnson Space Center) for the evaluation of complex workstations vis-a-vis the normal and emergency procedures they are intended to support and the types and number of individuals who must carry them out (1).

The short-term goal of the TEMPUS project is to allow a user to model a 3-D workstation environment, a set of human figures, and the tasks those people are to carry out. Those tasks are interactively specified in terms of body positions, reaches, and views relative to any of various coordinate reference frames. If the user specifies a task for the agents to carry out, TEMPUS will compute their required movements according to the current agent size and joint limits, report on whether they lead to collision conflicts, then display an animation of the activity for the user's visual evaluation. Since the user can alter the workstation, task, and agent characteristics, TEMPUS will assist users to design and evaluate workstations in response to a wide range of conditions (1).

Some of the major features of TEMPUS which, when taken in conjunction, make it a unique computer graphics application system include capabilities to:

* Create one or more male or female, (space-) suited or "shirt-sleeved" human figures which are correctly scaled according to a specific population, or which meet certain size constraints.
* View the human figure in any of several graphical modes: stick figure, line or shaded polygons, or shaded BUBBLEPERSON.
* Position the figure in any admissible position within joint angle constraints, and with the assistance of a robotics reach positioning algorithm for limbs.
* Combine the figures with three-dimensional polyhedral objects derived from an existing CAD system.
* Create shaded color graphics images of bodies in such environments.
* Use all TEMPUS features in an extensible and uniform user-friendly interactive system which does not require any explicit programming knowledge.

There are a number of other computer graphics systems for the modeling of human figures (2), but each of them fails to have one or more of the TEMPUS features listed above. Notable systems include Combiman (3), CYBERMAN (4), SAMMIE (5), NUDES2 (6), Skeleton Animation System (7), BBOP (8), LABAN (9), and others (10,11). Most of these fail to provide integral shaded graphics; others fail to provide convenient anthropometric selection and variety; still others lack convenient user interfaces or portable graphics.

TEMPUS evolved from previous work in human body modeling at the University of Pennsylvania (12,13,14,15,16). The relative efficiency of the uniform shaded sphere representation for human body graphical display led to its natural application in an engineering and human factors

environment. In June 1982 work began on TEMPUS: a NASA-funded extension to the older BUBBLEMAN system. Its design was based on the principal features listed above, with a target delivery date of June 1983. Undaunted by (or perhaps unaware of) the eventual scale of the project, a number of students began working on various components of the system in parallel, achieving system integration through clean communication between carefully designed modular subsystems.

Software engineering for TEMPUS included software conversions, SIGGRAPH CORE graphics use (17), and careful device interface design. Existing code from a Univac 90/70 and Univac 1100 implementation of BUBBLEMAN was converted to VAX-11/780 under VMS. Fortunately all necessary code was in relatively transportable PASCAL and FORTRAN. All graphics for TEMPUS (except the shaded image generation) became mediated entirely by a CORE graphics system written in PASCAL, also converted from the Univac 1100 (18). (Significant effort has been expended to increase the scope of the implementation since that report.) Among other features, the CORE enforces a strong separation between the device dependent and device independent parts of the system. Consequently, TEMPUS can be easily installed on other graphical display devices. In particular, the development system at the University of Pennsylvania uses a Grinnell frame buffer while the NASA JSC implementation runs on a Lexidata 3400 Solidview system. All graphics dependencies are neatly buried in the FORTRAN device "drivers."

The actual application-dependent code was broken down into several parallel tasks (Figure 1):

* The high level control structure for the interactive system.
* The menu subsystem and screen viewport utilization.
* The macro instruction and interaction archiving subsystem.
* Body sizing code based on anthropometric data.
* Novel reach positioning algorithms.
* Body model development.
* A hierarchical polyhedral object modeling system.
* Geometric algorithm support.
* Shaded graphics generation.
* CORE software development.

All the pieces came together in a July 1983 first installation of the system at NASA Johnson Space Center. The second installation came in October 1983, and the third in May 1984. Most of the original goals of the TEMPUS system were met. Many additional components of TEMPUS are under development and will be sketched in the final section.

SPECIFYING BODY SIZES

Human figures used within TEMPUS may be specified in a number of ways. They may be sized by specifying a specific individual in a personnel database, by supplying body segment lengths, or by fixing the global or individual segment percentiles of a given statistical population. Most of the body sizing information has been gathered from published regression formulas used in the CAR reach analysis system (19). Figure 2 illustrates a set of three individuals which correspond to 5th, 50th, and 95th percentile bodies computed from a database of Space Shuttle crewmembers. These are not simply scaled versions of one figure; the sizes are anthropometrically derived. Each figure (though shown here in the same position) is independently and concurrently manipulable. (The polygon "polybody" figures are abstractions of the human body, but are used to convey the obvious changes in segment size and proportion.)

Some of the unique features of the anthropometric capability in TEMPUS are:

* Multiple body data bases are allowed.
* The statistics used for percentile determination are based on the current data base and updated with that data base. Other systems, such as COMBIMAN (3), use statistics tables gathered from archival sources and thus do not as accurately reflect the statistics of a specific (and changing) population as we are required to do.
* The user can select members of the data base or create "generic" individuals satisfying certain size or percentile conditions or restrictions.

These choices allow variation of the anthropometric characteristics of the agents, thereby affecting things like arm reach, step size, clearance, joint limits, etc., and hence how they can carry out a particular action. For example, if an agent is required to turn a dial with his right hand, his right hand must be "at" the dial. If not already there, the agent may be able to get there just by stretching out his right arm, although a smaller agent might need to bend at the waist, or even move his entire body (say, by walking) over to the dial as well. All the joint angles of many reach actions need not be detailed by the TEMPUS user, since a reach positioning algorithm that operates on any body and body limb is available (described below).

3-D WORKSTATIONS

Since people work in a physical environment, we must be able to specify the objects in the workstation (levers, dials, gauges, tables, chairs, etc.) and their physical configuration. Workstations consist of object primitives in a "part-of" hierarchy typically derived from independent functional components, with a variety of attributes relating to their visual characteristics (color, surface material, and possible motion, for example). In the present application, the source of some of this information is an extensive data base of Shuttle components, payloads, and workstations created by the Crew Station Design Section of JSC using a Computer-Aided Design system called PLAID (1). Data conversion routines convert PLAID objects into the required TEMPUS polyhedral representational system and vice versa. (An IGES (20) interface was considered but has been tabled until adequate topology specifications are standardized.) One conversion is used when shaded graphics renderings of PLAID workstations are required; the other conversion is primarily used to model suited crewmembers. Given a particular individual, the suit parts are automatically selected, sized to fit, and then inserted back into the PLAID data base.

The internal polyhedral modeling system used to create and manipulate objects in TEMPUS is a locally designed and implemented geometric modeling package called SurfsUP ("Surface System of the University of Pennsylvania"). SurfsUP represents objects as trees, which denote part-of hierarchies. Attributes such as color, glossiness, visibility, transparency and highlighting can be placed on attribute lists located at each node in a hierarchy. An attribute is inherited downward but can be overridden by attaching the same attribute to a node lower in the hierarchy. Attributes and transformations can be either constant or variable.

There are three types of nodes that can have children. A branch node is simply a node with a list of children. A transformation node relates the coordinate system of its single child to the coordinate system of its parent. (Homogeneous matrices are currently used to specify the transformations but provision is made for other

methods such as Euler angles or an axis and angle. The latter method is more convenient when modeling rotating parts.) This type of node provides relative motion capabilities like that of GMSolid (21). An instantiation node provides variable instantiations that are inherited downwards.

The rest of the node types are for nodes that have no children, called "primitive" nodes. Among the node types are polyhedral surfaces (PSURFs), people, cameras and light sources.

PSURFs can be used to model solid objects (using a boundary representation) but can represent other topological structures as well. For generality, a winged edge data structure is not used; instead there is an explicit record for each point, edge and face, and attributes can be placed on any of these.

IntSurf is an interactive command driven workbench "front end" for SurfsUP. It allows access to procedures and data structures at a lower level than available through the TEMPUS program. IntSurf uses alphanumeric command strings and a simple postfix syntax. It is mainly used for testing and debugging, and for creating workstations that can later be read into TEMPUS.

POSITIONING TEMPUS OBJECTS

One of the major benefits of embedding most TEMPUS objects in the object hierarchy described above is the uniformity with which movement commands may be designed. TEMPUS cameras, light sources, workstation parts, and whole people are all positioned with the same commands, and all of these commands are implemented by modifying the moved object's parent transformation node.

Movements may be done with regard to any object's coordinate system. When discussing a movement, then, there is always the "moved object" and the "reference frame object." The moved object is always adjusted with regard to the coordinate system of the reference frame object; in this way the relationships between the two objects can always be explicitly displayed to and adjusted by the user. An object may also be its own reference frame, allowing a simple means of moving an object with regard to its own coordinate axes. In addition TEMPUS allows the user to create abstract "coordinate system objects" which may be moved and then used as reference frames.

Four types of positioning are available: translation, orientation, rotation about an object, and facing. Translation is the simplest: the moved object is simply translated along the axes of the reference frame object.

Orientation specifies the relative orientation of the moved object with regard to the reference frame object, independent of any translation between the origins of the two objects' coordinate systems. The three rotational degrees of freedom are parameterized in the following manner: given a chosen moved object axis, its orientation relative to the corresponding reference frame axis is determined by two spherical angles (longitude and latitude measures) and a final rotation about that chosen moved body axis. This parameterization results from three consecutive rotations about the moved body's axes and was found to give the user a better intuitive feel for the effects of rotations than if three rotations about the fixed reference frame axes were used.

A separate command is used to rotate the moved object about the reference frame object. Here, the relative orientations of the two objects are irrelevant: only the position of the origin of the moved object relative to the coordinate axes of the reference frame matters. Again, spherical coordinates are used for positioning, giving the user the longitude and latitude of the origin of the moved object relative to the reference frame coordinate system. These two degrees of freedom may later be supplemented by a rotation about the axis connecting the origins of the two objects and a translation along that axis.

The final positioning command, the facing command, allows the user to aim an axis of the moved object toward the reference frame origin. This is essentially a change in orientation without requiring "mental" rotations on the part of the user.

These commands are important for all object movements, but they are especially suited to changing the user's view of the TEMPUS world model. TEMPUS "cameras" determine the CORE viewing parameters which in turn control what is displayed to the user. The coordinate system of the current camera is always maintained, and may be positioned just as any other object with the flexibility allowed by the commands discussed above. In particular, it is especially convenient to aim the camera at an object with the facing command and then investigate various views of that object by rotating about it. The camera itself may also be used as a reference frame, a feature which is often useful.

A HUMAN MOVEMENT SIMULATOR

Our previous work in representation of human movement (14,22,23) led to the design and partial implementation of a human movement simulator that would execute motion descriptions (15). Among the primitive actions executable by the simulator were rotations, goal-directed positions or "reaches," and facings. (The other primitives are paths of motion (24,25) or "shapes," and contact relationships between two or more bodies, body parts or the environment. The former is to be specified in a forthcoming animation subsystem; the latter may be viewed as the special case of reaching moving points.) Rotations and twists have rather straightforward implementations as changes to the joint angles, and thus transformation matrices, at the affected joints. The "reaches" take us into the robotics domain (26,27) and are essential to the convenient specification of human (or robot) motion (28).

Interactive commands generate a sequence of reach actions or body positioning instructions. Figure 3 shows a multiple exposure illustration of an arm reach to a specified cartesian goal point. The joint angles are computed by the algorithm described below, and are not simply interpolated. This may be seen from the changing elbow angle and the straight-line fingertip path. Reach actions are computed for the user to minimize joint angle specification. The user positions the arms or legs of a human figure by specifying an adjustment to the (X,Y,Z) spatial position of some end-effector, such as the fingertip, grip center, or wrist. The reach uses the three degrees of freedom of the shoulder and the single degree of freedom at the elbow to achieve the three positional constraint goal (X,Y,Z), so the system is singly redundant. Of particular importance is the requirement that joint limits not be violated. Joint limits for spherical joints (such as the shoulder) are modeled using spherical polygons. For example, the direction in which the upper arm points is constrained to lie within a spherical polygon on a sphere about the shoulder. Upper arm twist is modeled with independent limits.

The steps of the algorithm are as follows:

1. Determine the elbow angle required by the goal using the law of cosines, and check it against the elbow joint limits.

2. Obtain the vector function E(phi), which describes a circle on which the elbow is constrained to lie by the shoulder position and the goal position for the end effector.

3. Obtain the arc (or arcs) on the circle E(phi) to which the elbow is restricted by shoulder joint limits. Limits on all three joint variables of the shoulder must be accounted for. The first two (spherical angles) are handled by finding the portion of the elbow circle which lies within the spherical polygon describing joint limits. The third

(upper arm twist) is handled by obtaining the points on the elbow circle corresponding to the minimum and maximum twist values.

4. Obtain the domain of the elbow circle parameter phi which corresponds to the permissible arcs on the elbow circle.

5. Choose some value for phi from its domain and evaluate E(phi). An attempt is made to maintain continuity for phi, if joint limits allow. (phi may also be adjusted independently, to move the elbow while keeping the end effector fixed.)

6. Obtain the joint angles required to achieve the resulting elbow position.

This procedure is in the process of being extended to handle position-orientation goals. The method depends on a useful observation of human limb structure. The arm (leg) is modeled as a seven degree of freedom chain with a spherical joint at the shoulder (hip), revolute at the elbow (knee), and another spherical joint at the wrist (ankle). The goal now has six contraints, three from position and three from orientation of the hand (or foot).

In the position-only solution, the end effector (the wrist, for example) is positioned assuming that the torso was fixed. Now we must contend with the same joint limits as before, and an additional set at the wrist. These may be handled by simply turning the problem around and "positioning the shoulder," with the hand held fixed in its goal position. Thus, the position-orientation problem is solved by two applications of the position-only solution and some coordinate transformation "glue."

3-D COLOR GRAPHICS

TEMPUS is an anthropometric modeling system having complete three-dimensional modeling and display capabilities incorporating a solid shaded human figure and workstation environment. Shaded images of human figures are shown in Figures 4 and 5. The three-dimensional models (Figure 5), called BUBBLEPEOPLE (13), are a particularly efficient method of graphical modeling and display of a solid figure, and greatly facilitate the visualization of human motion over existing vector approaches (2,3,5). A space suited figure, however, is mechanical enough to model effectively with polygons (1).

The figure may have any colors associated with body spheres to simulate (very tight fitting) "shirt-sleeved" clothing. A face with fixed features is also modeled with spheres. The face is important to the viewer as the primary cue for assessing the facing direction of the figure.

The shaded graphics displays of SurfsUP polyhedral objects include the usual shading models, while the BUBBLEPEOPLE are rendered in their particular flat sphere shading (13). The current implementation supports translucent polygons, shadows (including "soft" edges) and multiple light sources (29). While many other graphics systems with translucency and shadows exist, a unique feature of TEMPUS is that they are supported through a modified z-buffer algorithm. The useful features of a z-buffer which led to this choice is its ability to handle spatially unsorted primitive surfaces (such as polygons), and the emergence of fast shaded polygon rendering hardware (such as the Lexidata Solidview, Raster Technologies 1/25S, and the Weicat Tiling Engine). Since at least one of the target installation systems was of this type (a Lexidata Solidview system), the z-buffer algorithm permitted a simpler interface across different raster display systems. The typical z-buffer implementation is extended by the addition of translucency effects without the usual pattterned (partial) tiling of polygons. Likewise, multiple light sources and soft shadows are managed in a different fashion than previously reported (30,31).

Raster images are produced in a modified depth-buffer which stores information necessary to determine visible surfaces and the shadowing of these surfaces. Each cell in the two-dimensional array represents a visible point that is described by a record. In addition to the standard z value needed to determine which portions of a scan converted (tiled) polygon or sphere are currently "frontmost" in the current view (32), the following fields are included in the record: object pointer, surface normal, Sf (front-facing shadow: near surface of shadow volume) pointer, Sb (back-facing shadow: far surface of shadow volume) pointer. The object pointer leads to a description of the polygon (sphere) that is visible at this cell. Geometry, color and material attributes of the frontmost object at a cell can be accessed through the object pointer. The normal stored at the cell is the normal to the polygon (possibly an interpolated value) that is computed during the tiling process.

In order to determine whether or not a point is in shadow, it is necessary to keep track of the shadow polygons that surround the point. If there are multiple light sources then a point may be enclosed by shadow volumes due to different light sources and therefore it is necessary to keep a list of shadow polygons that surround the point. An identifier is associated with each shadow polygon to indicate the light source that is being blocked. Also, a z value is associated with each shadow polygon. This is used during the intensity calculation to determine if a pixel is in shadow. The Sf pointer points to a list of front-facing shadow polygons and the Sb pointer points to a list of back-facing polygons.

Intensity values are NOT computed and stored during the tiling process which is why the object pointer and normal fields are necessary. Although these additional fields increase the size of the buffer structure, they allow for the handling of transparency and shadowing with a depth-buffer algorithm. To handle the increased size of the buffer structure, a "multiple buffer" method is employed. That is, the screen is divided into multiple buffer boxes (for example, on a 512x512 display, the buffer boxes are 64x64). The rendering and shading algorithms are repeated for each of the buffer boxes. When the intensity values of a buffer box have been computed, they are sent to the display.

ANIMATION

Animation commands are presently stored as "history lists" in a movement representation derived from our previous work (14,22) and may be saved as movement "macros" for future use or combination with other actions (33). Under development is a "multiple track" (34,35) animation subsystem where several figures, or actions for one figure, may be arranged in time, and task time durations changed. The tasks may be shuffled, executed in parallel, synchronized along a time line, or phrased for smoother motion. Any sequence of body stances may be interpolated (using smooth spline curves (8,36) to animate successive actions.

Appropriate "run-time" tests exist for monitored conditions, such as illegal collisions or interference (13), and the agent's ability to achieve an intended goal during a reach. We are aware that collision avoidance (37,38) is an important problem during reach planning, but are not considering it at this time, preferring instead to rely on user interaction and experimentation.

A USER-FRIENDLY INTERACTIVE SYSTEM

The TEMPUS user is presented with a color graphics display (Grinnell 512x512x24), a digitizer tablet (typically) with a four-button cursor, and an auxilliary alphanumeric display terminal. Almost all communication between the user and the TEMPUS system is initiated by menu or valuator selection from the color graphics screen (for example, Figures. 2, 3, 4, and 5) or (alternatively but equivalently) from the alphanumeric terminal. This arrangement

encourages experimentation and permits a user to access TEMPUS with little or no training period.

The interactive control mechanism is designed to meet certain goals to insure usability, extensibility, and portability:

* Simple command selection mechanism, though facilities were to be provided to improve interaction for more experienced users.
* Orthogonality of application program and graphical output.
* Communication, rather than total integration of system components.
* "Macro" facility to save interaction sequences for system restoration, archive, and repetition purposes.

The interactive system design selected for TEMPUS is based on the User Interface Management System concept (39,40) which cleanly separates the user from the application routines. Figure 6 shows a block diagram of the major code sections of a UIMS-based design.

TEMPUS commands exist in a relatively shallow hierarchy (41). To avoid constant movement up and down the command tree, three separate and mutually orthogonal structures are available: menu items, permanent menu items, and switches. Menu items provide access to the leaf nodes of the command tree where actions are actually invoked. An explicit trace of menu picks down the hierarchy is always displayed; moreover, these items are themselves pickable to rapid-transit elsewhere in the tree (Figures 2, 3, and 4). Permanent menu items can be selected at any point during interaction without necessarily loosing one's place in the hierarchy. For example, the macro facility is always available as a permanent menu item; also, the system control commands such as "exit" and "abort" are here. Switches control internal modes (usually of display characteristics (line/shaded graphics) or units of measure (feet/meters)) which have no direct bearing on the kinds of manipulations being performed. While interactive systems of this sort are more difficult to program initially, the transparency and consistency of the interaction throughout the entire TEMPUS system has been well worth the effort.

To reduce the number of selections, TEMPUS makes extensive and consistent use of the concept of "currents." Each user accessible data type has a current value which is the default for any action which requires a value of that type. For example, there is a "current body," a "current camera," a "current workstation part," and so on. The control structure is designed in such a way that the user can change the "current" values without leaving the menu hierarchy position, so that re-executing the leaf node action with the new current values is done simply by another pick. Repetitive actions are therefore much easier to perform than if the user had to back out of the command hierarchy and work down again with different arguments. The resulting context sensitivity is a powerful feature for the experienced user.

CONCLUSIONS

TEMPUS is a unique integration of anthropometrically-based human figure generation, robotics motion modeling, advanced 3-D graphical display techniques, and task animation. In this report we have briefly described several features of TEMPUS which make it a flexible, usable, and unique system for human figure modeling.

The TEMPUS project has proved to be an enormously educational experience for our research group. Not only has it focused our previous research efforts into a full-scale graphics system software engineering project, but it has provided numerous directions in which our current research may proceed.

ACKNOWLEDGMENTS

This investigation has been greatly aided by many people, including Steve Platt, Doug Bloom, Luke Weinstein, Bonnie Webber, Carolyn Brown, Frank Ganis, Cheryl McNalley, John Hagan, Jane Rovins, Richard Duncan, David Turvene, Mike Mahoney, and Jim Jeletic. The BUBBLEWOMAN picture was created by Robin Pyle and Marion Hamermesh. This research is supported by NASA Contracts NAS9-16634 and NAS9-17239, NSF CER award MCS-82-19196, and ARO Grant DAAG29-84-K-0061.

REFERENCES

(1) Brown, J. W., "Using computer graphics to enhance astronaut and systems safety," Proc. 15th Symposium on Space Safety and Rescue, International Academy of Astronautics, 33rd International Astronautical Federation Congress, Paris, France, 1982, pp. 1-8.

(2) Dooley, M., "Anthropometric modeling programs -- A survey," IEEE Computer Graphics and Applications 2(9), Nov. 1982, pp. 17-25.

(3) Bapu, P., Evans, S., Kitka, P., Korna, M., and McDaniel, J., User's guide for COMBIMAN programs, Univ. of Dayton Research Institute, U.S.A.F. Report No. AFAMRL-TR-80-91, Jan. 1981.

(4) Blakeley, F. M., "CYBERMAN," Chrysler Corp., Detroit, MI, June 1980.

(5) Kingsley, E., Schofield, N., and Case, K., "SAMMIE-a computer aid for man-machine modeling," Computer Graphics 15(3), Aug. 1981, pp. 163-169.

(6) Herbison-Evans, D., "NUDES2: A Numeric Utility Displaying Ellipsoid Solids," Computer Graphics 12(3), Aug. 1978, pp. 354-356.

(7) Zeltzer, D., "Motor control techniques for figure animation," IEEE Computer Graphics and Applications 2(9), Nov. 1982, pp. 53-59.

(8) Williams, L., "BBOP," 3D Animation Seminar Notes, SIGGRAPH 1982.

(9) Calvert, T., Chapman, J., and Patla, A., "Aspects of the kinematic simulation of human movement," IEEE Computer Graphics and Applications 2(9), Nov. 1982, pp. 41-50.

(10) Fetter, W., "A progression of human figures simulated by computer graphics," IEEE Computer Graphics and Applications 2(9), Nov. 1982, pp. 9-13.

(11) Willmert, K. D., "Visualizing human body motion simulations," IEEE Computer Graphics and Applications 2(9), Nov. 1982, pp. 35-38.

(12) Badler, N., and Bajcsy, R., "Three-dimensional representations for computer graphics and computer vision," Computer Graphics 12(3), Aug. 1978, pp. 153-160.

(13) Badler, N., O'Rourke, J., and Toltzis, H., "A spherical representation of a human body for visualizing movement," IEEE Proc. 67(10), Oct. 1979, pp. 1397-1403.

(14) Badler, N., and Smoliar, S., "Digital representations of human movement," Computing Surveys 11(1), March 1979, pp. 19-38.

(15) Badler, N., O'Rourke, J., and Kaufman, B., "Special problems in human movement simulation," Computer Graphics 14(3), July 1980, pp. 189-197.

(16) O'Rourke, J. and Badler, N., "Model-based image analysis of human motion using constraint propagation," IEEE Trans. PAMI 2(6), Nov. 1980, pp. 522-536.

(17) Graphic Standards Planning Committee Report, Computer Graphics 13(3), Aug. 1979.

(18) Stluka, F., Saunders, B., Slayton, P., and Badler, N., "Overview of the University of Pennsylvania CORE system standard graphics package implementation," Computer Graphics 16(2), June 1982, pp. 177-186.

(19) Harris, R., Bennet, J., and Dow, L., "CAR-II - A revised model for crew assessment of reach," Technical report 1400.06B, Analytics, Willow Grove PA, June 1980.

(20) Smith, B., Brauner, K., Kennicott, P., Liewald, M., and Wellington, J., "Initial Graphics Exchange Specification, Version 2.0," PB83-137448, NTIS, Springfield, VA.

(21) Tilove, R., "Extending solid modeling systems for mechanism design and kinematic simulation," IEEE Computer Graphics and Applications 3(3), May/June 1983, pp. 9-19.

(22) Weber, L., Smoliar, S., and Badler, N., "An architecture for the simulation of human movement," Proc. ACM Annual Conf. 1978, pp. 737-745.

(23) Badler, N., "Design of a human movement representation incorporating dynamics and task simulation," 3D Animation Seminar Notes, SIGGRAPH 1982.

(24) Shelley, K. and Greenberg, D., "Path specification and path coherance," Computer Graphics 16(3), July 1982, pp. 157-166.

(25) Loomis, J., Poizner, H., Bellugi, U., Blakemore, A., Hollarbach, J., "Computer graphic modeling of American Sign Language," Computer Graphics 17(3), July 1983, pp. 105-114.

(26) Korein, J., and Badler, N., "Techniques for goal directed motion," IEEE Computer Graphics and Applications 2(9), Nov. 1982, pp. 71-81.

(27) Korein, J., A Geometric Investigation of Reach, MIT Press, 1985.

(28) Paul, R., Robot Manipulators: Mathematics, Programming, and Control, MIT Press, Cambridge, MA, 1981.

(29) Brotman,L.S. and Badler, N., "Generating soft shadows with a depth buffer algorithm," IEEE Computer Graphics and Applications 4(10), October 1984, pp. 5-12.

(30) Williams, L., "Casting curved shadows on curved surfaces," Computer Graphics 12(3), July 1978, pp. 270-274.

(31) Crow, F., "Shadow algorithms for computer graphics", Computer Graphics 11(2) 1977, pp. 242-248.

(32) Foley, J., and van Dam, A., Fundamentals of Interactive Computer Graphics. Addison-Wesley, Reading, MA, 1982.

(33) Bloom, D., "A user-oriented interface control of an interactive computer graphics system," MSE Thesis, Dept. of Computer and Information Science, Univ. of Pennsylvania, May 1983.

(34) Feiner, S., Salesin, D., and Banchoff, T., "Dial: A diagrammatic animation language," IEEE Computer Graphics and Applications 2(7), Sept. 1982, pp. 43-54.

(35) Fortin, D., Lamy, J.-F., and Thalmann, D., "A multiple track animator system for motion synchronization," to appear in Proc. ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion, April 1983.

(36) Kovacs, W., "Tools for motion design," 3D Animation Seminar Notes, SIGGRAPH 1982.

(37) Lozano-Perez, T., and Wesley, M., "An algorithm for planning collision-free paths among polyhedral obstacles," Comm. of the ACM 22(10), Oct. 1979, pp.560-570.

(38) Brooks, R., "Solving the find-path problem by good representation of free space," Proc. AAAI National Conf. on Artificial Intelligence, Pittsburgh, PA, 1982, pp. 381-386.

(39) Kasik, D., "A user interface management system," Computer Graphics 16(3), July 1982, pp. 99-106.

(40) Thomas, J. and Hamlin, G. (eds), "Graphical Input Interaction Technique (GIIT)," Workshop summary, Computer Graphics 17(1), Jan. 1983.

(41) Weinstein, L., "A menu driven user interface," MSE Thesis, Dept. of Computer and Information Science, Univ. of Pennsylvania, May 1983.

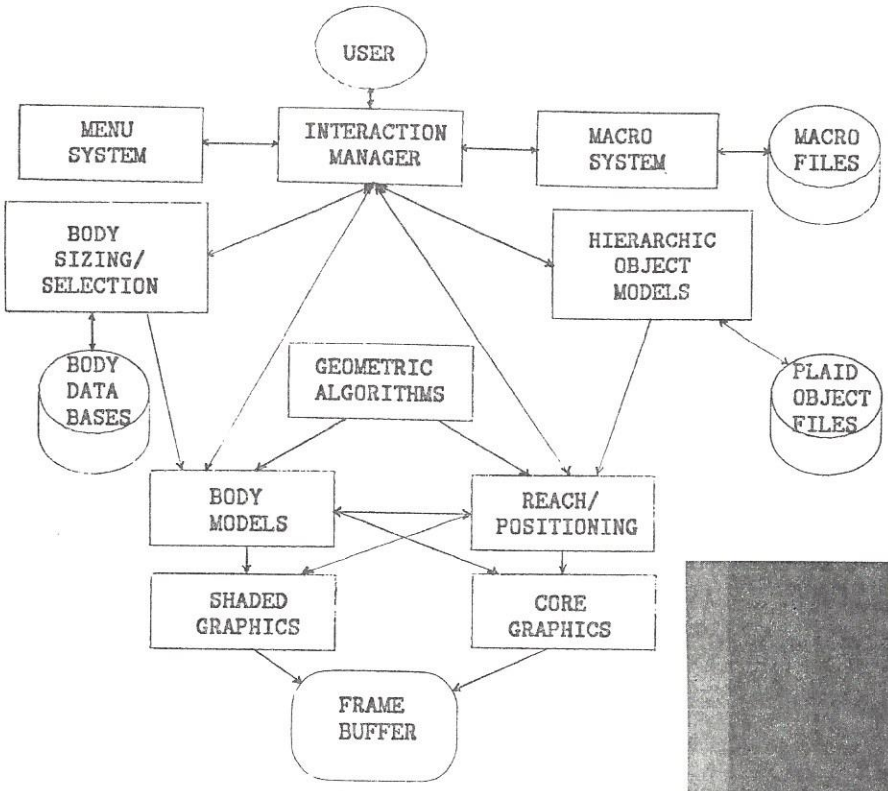

Figure 1. Block diagram of the TEMPUS system.

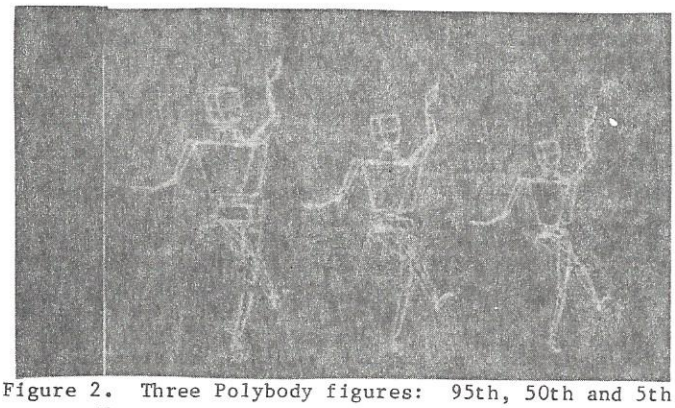

Figure 2. Three Polybody figures: 95th, 50th and 5th percentiles.

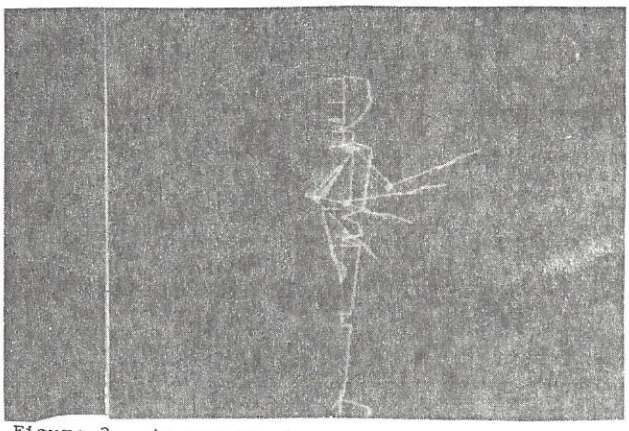

Figure 3. An arm reach

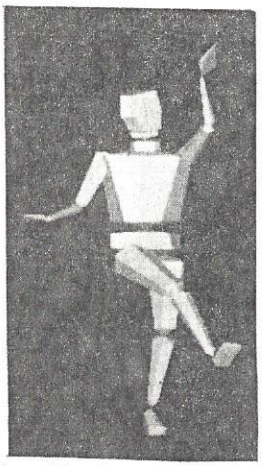

Figure 4. Simple shaded view of Polybody figure.

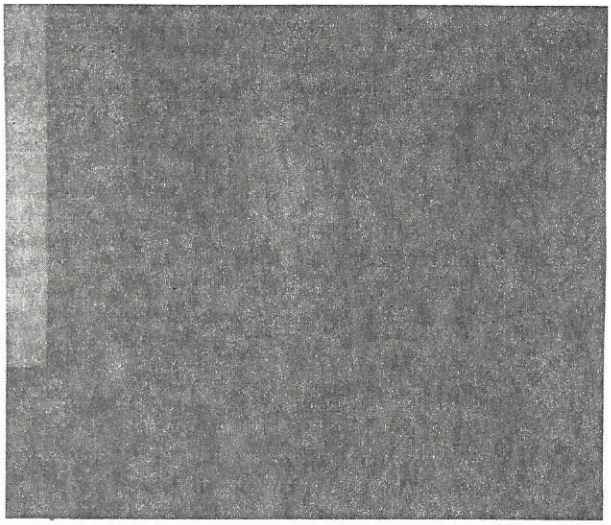

Figure 5. BUBBLEWOMAN shaded figure in a three-dimensional environment.

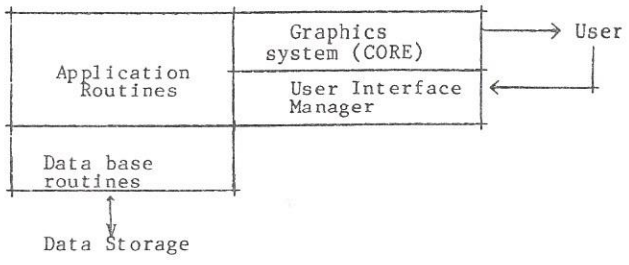

Figure 6. A User Interface Management System.