375

324-61

233713

N90-20675

Using an Instrumented Manikin for Space Station Freedom Analysis

Linda Orr, NASA/Johnson Space Center
Richard Hill, Lockheed Engineering and Sciences Corporation

One of the most intriguing and complex areas of current computer graphics research is animating human figures to behave in a realistic manner. Believable, accurate human models are desirable for many everyday uses including industrial and architectural design, medical applications, and human factors evaluations. For zero-gravity (0-g) spacecraft design and mission planning scenarios, they are particularly valuable since 0-g conditions are difficult to simulate in a one-gravity Earth environment.

At NASA/JSC, an in-house human modeling package called PLAID is currently being used to produce animations for human factors evaluations of Space Station Freedom design issues. This paper will present an introductory background discussion of problems encountered in existing techniques for animating human models and how an instrumented manikin can help improve the realism of these models.

BACKGROUND

The difficulty in creating realistic models of people lies in the complexity of the human body. There are over 200 degrees of freedom in the body structure [6]. For purposes of human modeling for task planning and motion studies, the body can be graphically represented as a series of rigid body joints and linkages. For many movements the human model can be adequately represented by a subset of 30-40 degrees of freedom if it is not necessary to model each finger, toe, spinal disc, etc. for a study [4]. Even with such simplification of body structure, however, the approach to animating human movement in a realistic manner remains a complex issue. With 30-40 degrees of freedom in a model, redundant solutions for a desired motion are possible, some of which may be more comfortable and intuitive for a human to perform than others are. (Fig. 1.)

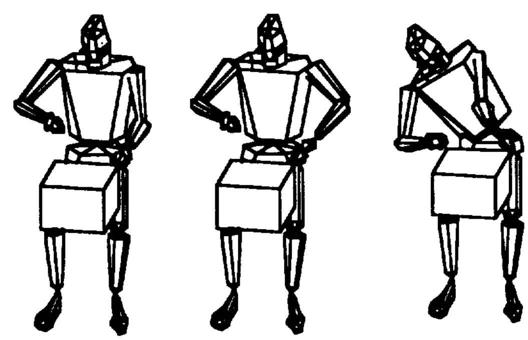


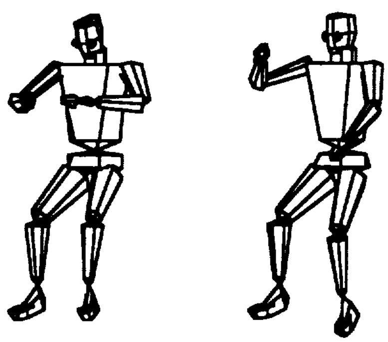
FIGURE 1

Redundant solutions for left hand reach with fixed feet locations.

There are basically three methods of modeling human motion for animated graphics display output: a guiding (keyframe) system, a program level or algorithm-based system, and a task level system [7]. Each method has its strengths and weaknesses.

Method 1: Guiding System

The guiding system is the traditional tool of computer animators dealing with human motion. Under this system, a user sets up a series of "keyframes" explicitly describing key actions of interest. For example, in Fig. 2a, a crewmember is modeled in an initial position configuration at time t_0. At time t_x, he/she has assumed a new position configuration of interest (Fig. 2b). The program is then instructed to calculate a number of frames (n) showing the in-between frames from t_o to t_x, usually at a rate of 30 frames per second for video output. The program can use a simple linear interpolation to compute the new position of link L at each frame between t_o and t_x, based on the distance of travel of link L during that time interval. Linear interpolation tends to make the motion of the figure appear jerky and unnatural, however. The motion can be given a smoother appearance by using a spline intepolation instead of a linear assumption.



(a) (b)

FIGURE 2

Crewmember at initial time to and later time tx.

Having the computer calculate in-between frames and check joint limits for solution feasibility can help the user relieve some of the tedium involved in animating human figures. This approach can be satisfactory when simple motion is all that is required. Where subtle changes of motion are desired, however, guiding systems require a lot of manual set-up time since they require more keyframes to define fully the action of interest. Much iteration is usually required to "tweak" the motion for it to look correct to a viewer. The motion generated is therefore highly dependent on the powers of observation of the animator.

The guiding method is particularly time-consuming to set up for three-dimensional animated studies since perspective views of the models and their work environments can be misleading. For graphically directing motion from one specific point to another, some guiding system users turn perspective off and look at 2-dimensional views for better precision in positioning body segments. This approach requires a view change to locate the third dimensional coordinate, and a decomposition of the movement into two or three orthogonal rotations, depending on the joint being manipulated. The view change and mental decomposition require additional set-up time.

Method 2: Algorithm-based System

In an algorithm-based system, physical laws are applied to human parameters. Typically, these systems assume rigid body mechanical links with joints modeled as spring and damper systems. The most commonly used algorithms are direct/inverse kinematics and direct/inverse dynamics algorithms borrowed from robotics applications.

The direct kinematics approach can be described as: given a set of joint angle information, determine the position and orientation of an end effector such as a hand or foot. Once position and orientation are determined, they can be differentiated to obtain joint velocities and accelerations. A simple example of a direct kinematics algorithm is the Denavit-Hartenberg matrix method [2]. The inverse kinematics problem is to determine appropriate joint angles given position and orientation of a desired end effector, and an example of such an algorithm is one described by Hollerback and Sahar [3].

The inverse kinematics approach is useful in reach evaluations for human factors studies. Given information on lengths of body segments, such algorithms can determine if Crewmember A at location (x,y,z) can reach button B without requiring the system user to predetermine (or guess) the desired joint angles. Since human beings have joint limits that restrict some motions, a good human modeling program will check joint limits for each frame of animation. Joint limit checking improves the animation result by eliminating solutions that are not humanly feasible to perform. The problem with joint limit checking is that it tells you nothing about the "naturalness" of the motions.

For dynamics analyses, the direct dynamics problem is described as determining the trajectories of the end effector(s) given appropriate initial conditions of force and torque parameters. The inverse dynamics solution is to determine the initial forces and torques on joints required to produce known resultant forces and torques at time t_x. For human modeling, the direct/indirect dynamics algorithms borrow heavily from robotics applications. The most commonly used dynamics algorithms generally fall into one of two categories [4]: Lagrange's equations of motion based on kinetic and potential energies for nonconservative systems, and Newton-Euler formulations based on Newton's second law for determining the total force vector and Euler's equation for determining the total torque vector.

A major drawback to modeling human motion with algorithms is that human motion is not purely kinematics or purely dynamics: it is a combination of both [1]. Dynamics simulations should produce accurate motion animations if the dynamic model is sufficiently detailed. Often, however, technically feasible but unnatural looking solutions are a result of dynamic modeling since it is difficult to come up with enough equations of motion, constraints, etc. to eliminate redundant solutions.

An additional problem with dynamics modeling of humans is that spring and damper functions, not constants, are required to describe humans accurately with spring/damper analogies. Determining these functions requires collection, storage and reduction of empirical data and such data is generally not available. Data supplied from cadaver studies can be of questionable value when applied to simulations of living people. Existing data from live subjects is usually limited to studies of specific motions or tasks and may not be universally applicable to all motion situations.

For realistic-looking animations based on algorithms, information may also be needed on motion comfort levels and preferred motion. For example, to retrieve an object dropped on the floor, does someone simply bend straight-legged from the hips or does he/she bend the knees and stoop part way? The result is that even with a reasonably detailed algorithmic model, the system user is still required to tweak the model to make its motion appear more natural to a viewer.

Method 3: Task Level System

This method uses Artificial Intelligence (AI) techniques to describe the performance of a task at multiple levels. For animation purposes, this requires applying a set of facts to rules about task actions. For a given task, high level AI commands, rules and descriptions of actions are used to describe the behavior of the human model in terms of events and relationships. The high level AI system transforms the behavior model into low-level instructions such as algorithm references or key values for parametric keyframe creation; these low-level instructions are then used to create an animation of the task performance [4,7].

Successful task performance interpretation requires knowledge of the task environment and objects within it. This knowledge usually involves an object oriented database that not only contains information about an object's geometry and mass attributes (e.g., density, specularity, thermal properties) but also how it is put together, how it behaves and whether it inherits properties from related objects. An example high level task command might be, "Put the book on the table." A task performance system must contain rules defining how the verb "put" is translated into a human motion, object information such as book dimensions and table height, information regarding which person is to put the book on the table, and the current state of the animation environment (Must someone first pick up the book or is he/she already holding it? Is the person close enough to use a simple arm reach to place the book on the table or must he/she walk across a room to complete the task?). A more sophisticated system could also check an anthropometric database for information about the individual performing the task to determine arm length and strength factors that might affect the task outcome.

Sophisticated task performance systems will take many years to develop. Rules for task performance must be created and iterated to perfect; knowledge-based object descriptions must be input to a database so the system can access the information needed for task simulation. The lengthy development time for perfecting task performance behavior rules and the problems of organizing the large database required for such a system are its chief drawbacks.

DISCUSSION OF MANIKIN DEVELOPMENT

Each of the three animation methods mentioned has strengths and weaknesses. At present, the authors see the PLAID human modeling effort eventually evolving into a program with heavy emphasis on task performance and algorithm-based methods with a guiding system user option. However, such a sophisticated modeling program will take years to develop. In the meantime, PLAID animators use a combination of guiding and kinematic algorithm methods to evaluate human factors issues for the Space Station Freedom Program.

Reach algorithms and joint limit checking are an integral part of PLAID's anthropometrics features but still require a large amount of user set-up time for some motion studies. The reach algorithm works quite efficiently when used to evaluate simple reaches to a predefined vertex on a person or object. A significant area of difficulty arose during some complex reach studies for the NASA Man-Systems Integration Standards (MSIS) document [5], however.

The MSIS is a 4-volume set of man-systems integration design considerations and requirements for development of manned spacecraft. Volume IV is specifically dedicated to Space Station Freedom human factors design issues. PLAID anthropometric features were used in the MSIS to help determine maximum reach envelopes of 5th percentile female and 95th percentile male astronaut candidates. For simple reaches, the existing PLAID features were straightforward to set up and manipulate. (Fig. 3). User set-up of imaginary 0-g maximum side-reach envelopes in free space with a foot restraint presented significant complications, however.

In Figure 4, the human model is initially positioned in a 0-g configuration with arms reaching above the head as far as possible and feet restrained in a foot restraint. The model is then positioned to sweep out an envelope in his/her lateral plane and identify points on that envelope. This motion is quite complex and eventually involves waist and/or hip twist, knee flexion, ankle flexion, etc. Since points on the envelope are in free space and are unknown to the system user, the reach algorithm (which requires a known destination vertex on a person or object) cannot be used. The user must therefore manipulate the various degrees of freedom on a joint by joint basis. Altering one joint affects the links downstream from it so the process is tediously iterative. Since the figure was being viewed on a computer screen, an inherently 2-dimensional display device, the user was required to make frequent view changes to ensure he/she understood exactly how the human model was currently positioned. For additional studies of complex motion a more user-friendly set-up procedure is obviously needed.

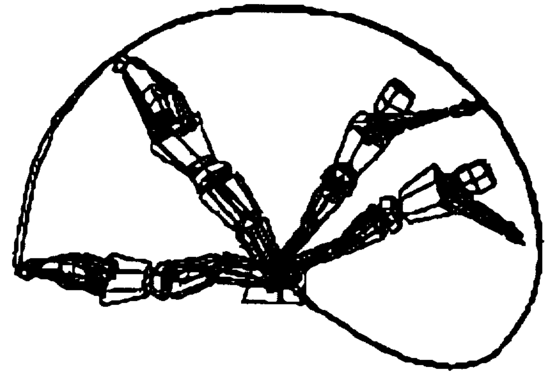


FIGURE 3

Simple forward/backward reach envelope with foot restraint for MSIS document.

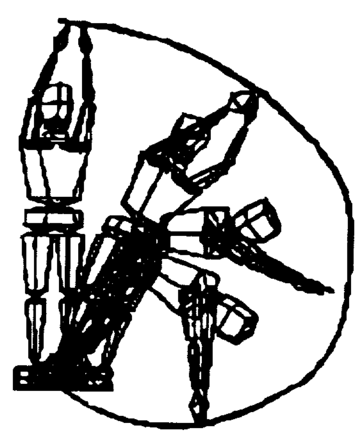


FIGURE 4

Complex side-reach envelope with foot restraint for the MSIS document.

A faster, more intuitive input device for positioning complex human movements in free space is an instrumented manikin. Such a device is currently being developed by the Graphics Analysis Facility at JSC for use with PLAID human modeling features. The manikin is a modified crash dummy with wirewound linear potentiometers instead of accelerometers for its instrumentation. It is approximately 48 inches tall and has 38 measurable degrees of freedom. To model actual human movement capabilities more closely, the standard crash dummy mechanical structure was modified to provide shoulder and thigh twist and was given a flexible neck.

The manikin is a truly 3-dimensional input device that can provide the computer with multiple position and orientation inputs simultaneously. It can be manipulated by a user in hands-on fashion to a desired configuration, where friction in the joints retains their positions once the user lets go. Alternatively, set screws can be used to lock the joints if preferred; for example, the user may want the legs configured in a 0-g orientation for an entire study. Mechanical joint limit stops equivalent to or slightly exceeding normal human limits are built into the structure.

The manikin is initially placed in a 1-g standing position and calibrated. When the user has manipulated the manikin to a new configuration, relative displacement voltages undergo an AC/DC conversion and signals are sent through an RS232 interface to the computer program. The input is converted to degrees for segment displacement information and then joint limits are checked by software to ensure position validity. Since PLAID body segment lengths are normalized, they can be read if desired from a user specified database of astronaut applicant data compiled by the Johnson Space Center's Anthropometrics and Biomechanics Laboratory. Thus, the manikin can be used to manipulate positions of different sized human models without mechanical or electrical reconfiguration.

CONCLUSION

By using the instrumented manikin, a user has a combination of algorithm and guiding methods available for setting up the desired study parameters. The user can utilize the power of algorithms as much as possible to simplify set-up procedures, yet have an effective way to tweak the human model for creating complex, subtle motion keyframes.

As a long-term animation system goal, an AI-based task performance system with heavy reliance on efficient algorithms is anticipated. While this system is being developed, however, human modeling analysts still need an effective tool to blend the individual strengths of guiding and algorithm methods. Even when the long-term system is in place, users will probably continue to demand an efficient way to modify the motion analysis output if desired. The instrumented manikin can be an effective tool for providing this option.

REFERENCES

1. Badler, Norman I. "A representation for natural human movement"; Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1986.

2. Denavit, J. and Hartenberg, R.S. "A Kinematic Notation for Lower Pair Mechanisms Based on Matrices", JOURNAL OF APPLIED MECHANICS, Vol. 22, 1955.

3. Hollerbach, J.M. and Sahar, G. "Wrist-Partitioned, Inverse Kinematic Accelerations and Manipulator Dynamics", INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH, Vol. 2, No. 4, 1983.

4. Magnenat-Thalmann, N. and Thalmann, D. (1988) "Course Notes on Synthetic Actors", SIGGRAPH '88, Atlanta, Georgia, 1988.

5. NASA-STD-3000, Man-Systems Integration Standards.

6. Zeltzer, D. "Motor Control techniques for figure animation", IEEE COMPUTER GRAPHICS APPLICATIONS, November 1982.

7. Zeltzer, D. "Toward an integrated view of 3-D computer animation", THE VISUAL COMPUTER: THE INTERNATIONAL JOURNAL OF COMPUTER GRAPHICS 1,4 (1985).