# GOAL-DIRECTED MOVEMENT SIMULATION

C. Csuri

*Ohio State University*

## ABSTRACT

The primary focus in this paper is upon techniques to facilitate motion control of complex structures such as skeletons in a three dimensional environment. Some of the developments in computer animation are briefly reviewed to provide a context for our animation system. Results from Physiology and Robotics are also examined. What three dimensional animation means to the artist or animator is considered as well as the value of real-time response and feedback. The concept of goal-directed movement simulation was chosen to suggest a synthesis of techniques used in robotics and artificial intelligence. This approach to an animation system involves the hierarchical control of functional synergies inherently goal-directed at the top level. Moreover, if decomposition into motion primitives, i.e., the synergies, is done correctly, the system will be trainable in the same sense in which a human or an animal can be taught a new motion sequence. Hence, such a system is extensible.

## RÉSUMÉ

Le but premier de la présente communication est de décrire des techniques facilitant la commande du mouvement de systèmes complexes tels que les squelettes, dans un espace tridimensionnel. On donne un court aperçu de certaines réalisations en animation sur ordinateur en guise de cadre de discussion pour notre système d'animation. Des résultats obtenus en psychologie et en robotique sont aussi examinés. Nous nous intéressons à ce que l'animation tridimensionnelle représente pour l'artiste ou le technicien en animation, ainsi qu'à l'importance d'une réponse ou d'une rétroaction en temps réel. Le concept de la simulation du mouvement "dans un but" a été retenu comme base de synthèse des techniques utilisées en robotique et dans les systèmes avec intelligence artificielle. Cette méthode d'animation repose sur la commande hiérarchique de synergies fonctionnelles qui sont implicitement dirigées "dans un but" au palier supérieur. En outre, si la décomposition en éléments fondamentaux de mouvement, c'est-à-dire en synergies, est réalisée de façon adéquate, le système peut "apprendre" de nouvelles séries de mouvements, à la façon de l'être humain ou des animaux. Ainsi, les possibilités d'un tel système peuvent être augmentées.

## 1. Introduction

The primary focus in this paper is upon techniques to facilitate motion control of complex structures such as skeletons in a three dimensional environment. We have chosen the concept of goal-directed movement simulation to suggest that we are attempting to synthesize techniques used in robotics and artificial intelligence. Our approach to an animation system, which we will discuss in detail later, involves the hierarchical control of functional synergies inherently goal-directed at the top level. A brief review of some of the developments in computer animation may provide a context for our evolving animation system. In addition, it may be useful to consider what three dimensional animation means to the artist or animator and the value of real-time response and feedback. Then we may be ready to examine some results from physiology and robotics, and discuss the design of the goal-directed movement simulation system.

One of the earliest computer animated films using key frame animation techniques was Hummingbird [8] (1967) and the technique was more fully realized in Real Time [9] (1970). The system developed by Wein [32] was used to produce the well-known film "Hunger" and there were other key-frame animation systems such as [2, 15]. These systems were useful aids to conventional animation but the transitions between key frames did not adequately convey complex three dimensional movement. Recently Reeves [27] has developed an approach for the inbetweening of key frames with the utilization of moving point constraints. Baecker's [2] ideas on P-curves were intended to provide more control for the animator and Reeve's work is an advance in the context of key frame animation. In general, many of the techniques for key frame animation are valuable aids to animation but all of these approaches have their own set of problems when dealing with highly complicated imagery. There are few tools to handle complex motion involving several figures with the cues for three dimensions. None of them address the issue of a knowledge-based system and there does not seem to be a concern about goal-directed movement simulation. In the final analysis, from the viewpoint of the animator, a system's utility can be measured by the time required to specify the movement and to achieve state of the art image quality.

In the three dimensional case the Computer Graphics Research Group's real-time display of a turtle (1969) was an early example of interactive techniques and animation. It was an attempt to graphically specify the necessary transformations. With a light pen the user had independent control of each of the appendages in real time. The same system enabled a user to "fly" a helicopter in real time, also with independent control of each of the rotors. However, they were more experiments than an animation system. Some interesting ideas were developed about real time and the importance of feedback in animation. Wessler [34] in his film, "Not Just Reality", used key frame techniques in three dimensions which illustrated different types of walks. This required a great deal of Wessler's effort, but the movement was still awkward. Badler [1] has approached the problem of complex motion in 3-D from the viewpoint of notation systems and he has attempted to employ certain aspects of artificial intelligence techniques in this work. Most notable is his work on a computerized version of Labanotation. This system for dance notation is interpreted from the Laban form and then translated into an animated display. His approach works well for an expert in Labanotation who can describe a dance sequence in 3-D but it seems to offer few tools for an animator to easily generate more general human movement. He does not seem to have solved the problem of portraying well-coordinated human movement.

An important objective of our research is to keep the animator's input to the animation system as simple and direct as possible. When we wish to display a figure walking across a room, we would like our script to be close to an English language description of the desired action. Thus the animation system must embody sufficient knowledge about walking and maneuvering in an environment to make this possible. The notion of "goal-direction" implies that the animation system is told WHAT to do, but not HOW to do it. Goal-direction, however, is implicit in many natural and artificial systems; an animation system based on the simulation of hierarchically organized, natural movement systems will be inherently goal-directed at the top level.

Moreover, we view motion specification as a major bottleneck preventing more practical techniques for animation. Although three dimensional computer animation also has difficulties, we are pursuing motion specification as a skeleton representation placed in the context of an environment because the

---

*This work is supported in part by National Science Foundation Grant No. MCS-7923670.

computer is well-suited for this task. If we continue with the motion specification problem, then this work can be interfaced to high quality three dimensional imagery. An animation sequence or file can be used to drive a set of routines for fleshing out surfaces to exploit the progress made in 3-D raster graphics. At The Ohio State University, with the work of Crow [7] in display algorithms and Carlson and Parent [4] in data generation, we have access to expertise and software to establish this relationship. While there has been significant progress in three dimensional raster graphics and animation of simple movement, advances in motion control of 3-D objects and scenes has lagged far behind. It is not enough to produce elegant geometric objects rotating in three dimensional space or to move along a path through a realistic terrain model.

## 2. Complex Animation

Computer modelling of subtle relationships that contribute to realistic human motion are exceedingly difficult to specify and control. The problem with human systems, in fact creatures in general, is that they exhibit great dynamic complexity, large dimensionality, non-linearity and variability of the kinematic structure. It seems evident that newer methods for computer graphics are required to easily specify complex motion.

Animators, in order to portray human locomotion in 3-D, must specify all joint positions using currently available computer graphics languages and techniques. Even with a higher level graphics language this process is time consuming and awkward, with the resulting animation looking like mechanical "wind-up toys." Furthermore, such animated displays are restricted to a walk or run on level ground, which is the easiest to imitate. (The running and timing associated with the track and field event of the steeple chase is far more difficult.) It is interesting to note that even with major productions such as Cinderella, the animator had serious problems trying to visualize complex scenes in 3-D. In fact, live performances had to be filmed so that complex timing and spatial relationships could be resolved.

It should be noted that the human body possesses some 200 degrees of freedom controlled by about 400 different muscles [24]. The problem of centralized, one-level control of so many degrees of freedom appears to be computationally intractable, even for biological systems, as shown by Bernstein [3]. Therefore, the key to understanding and controlling motion in complex motor systems lies in decomposing a large system with many degrees of freedom into a hierarchically coordinated set of smaller subsystems, each with few degrees of freedom.

The animator in this three dimensional universe tends to think of objects rather than lines. The artist moves an object in space (it is felt kinesthetically) rather than imitating that motion with lines and color. The same techniques used by a cinema director are inherent in such an animation system, e.g. pans, zooms, trucking shots, etc. While the end products produced by computers of three dimensional color creatures and environments could look like the work of Jiri Trinka, Willis O'Brien or Jim Hensen, the methods required for motion control are radically different.

The animator views the world or the representative data base more like a sculptor working "in the round" rather than a painter who selects a single viewpoint. Our notion of 3-D space, how it resides in our consciousness and possibly what visual cues triggers the perception, are topics somewhat beyond the scope of this paper. While at a theoretical level there is an epistemological problem about the validity of one's knowledge of 3-D space and reality, as a practical matter this new medium affects the way one perceives animation. As computer animators, we are conditioned by the way we must describe objects and scenes in three dimensions, and how in practice we position objects and generate three dimensional paths.

We are basing our construction on an analysis of natural movement systems--something every successful artist and animator has done. This will lead to generality in the kinds of motions the system will be capable of displaying. The principles of control of motion seem to be rather universal throughout the animal kingdom. Thus an animation system that embodies a correct analysis of motion control ought to be capable of generating the motions of all sorts of human and non-human figures. This will give the animator a great deal of artistic freedom in developing animation sequences, since he or she will be spared the tedium of concocting highly detailed motion specifications or long lists of rotations and translations, and instead can concentrate on the ideas, images, and effects to be portrayed.

## 3. Three Dimensional Computer Animation

Three dimensional computer animation may be conceived of as a synthesis of simulation and interactive computer graphics. We are searching for a computer model of skeletons in the general case. The study and analysis of the behavior of the original system provides us with knowledge to make applications to computer animation.

In many ways the proposed function of such a system parallels the functioning of a system designed to control a robot, or an anthropomorphic manipulator. In a graphics system the output of the movement controller would be a set of joint angle commands to some kind of animation processor. In the case of a robot or manipulator, the output would also consist of a set of joint angles, but in addition the robot controller must compute the actual torques or gains necessary to move the physical mass of the device. The animation movement controller is spared this burden--it need only ensure that the motion of the figure proceed from some start configuration to some desired configuration in a "natural" way.

In instances where computer animated displays have been produced of a skeleton-like stick figure the motion portrayed can be very realistic. Human gait movements are cyclical and easier to specify than the action required for a general animation system. People, however, pick up objects, jump awkwardly, twist, turn, hop, wrestle, fight and generally move in ways that are non-cyclical. What we seem to need is an approach that uses motion simulation techniques and one where the user can graphically specify the animation.

## 4. Real-Time Display and Preview

One of the problems is to design a system where the animator can interactively specify the motion required for an animation sequence. He should be provided with tools to graphically control the movement of complex structures. The animator should be able to create and manipulate realistic human or animation figures in a convenient and natural way. Another important requirement is the real-time feedback, editing, and smooth display of skeleton motion in three dimensions. While algorithms representing the physics of a moving skeleton are computionally intensive, special-purpose hardware can be built to overcome a number of limitations. We have a program task under our National Science Foundation grant to design and build such a processor using a bit-slice architecture. Of course, one can conceive of circumstances where the processor would soon be bounded computationally. While one skeleton may move in real time, it becomes a serious problem to move several skeletons through an environment. One could evolve an architecture with parallel processing schemes. However, as a practical matter we plan to use a real-time playback scheme to preview the motion files.

## 5. Results from Physiology and Robotics

A biological motor system constitutes an elegant evolutionary solution to an extremely complicated set of problems, namely, the task of maneuvering a multilink structure with many degrees of freedom through a dynamic environment. Indeed, the literature on the question of animal locomotion is punctuated with remarks on the extraordinary complexity of the process.

A large body of evidence suggests that natural movement systems have evolved into hierarchical control structures coordinating largely autonomous subsystems. [16, 26, 17] These subsystems are organized as "functional synergies", that is, modules consisting of a set of muscles and joints that can effect a particular class of motions, each module under the control of a set of "local motor programs" (LMPs) [13]. Many physiologists have come to view volitional movement as being "built on a base of reflex processes", or just such low-level motor programs coordinated and regulated by higher-level controllers [11]. As Peter Greene and others have argued, this may be the only economical way to control systems with very many degrees of freedom. [16, 18, 26]

Local motor programs have been found to underlie all kinds of stereotypical behavior in animals [26]. One such LMP that has been identified and studied in humans is the LMP for controlling posture. [13] LMPs for the control of walking have been extensively studied in cockroaches and cats [26]. These latter investigations have shown that the cyclic limb activity that results in walking is controlled by LMPs residing in nerve centers in the spinal cord. The brain, in turn, activates these spinal centers and sets global parameters such as the overall rate of motion. Rather than storing explicit movement descriptions, the motor system appears to maintain a repertory of local motor programs which can be combined in various ways to fit the circumstances at hand.

The coordination of LMPs has been characterized in the following way: 1) local motor subsystems function as autonomously as possible, with a minimum of communication with other subsystems; 2) local subsystems are goal-directed from above and respond to feedback from below; 3) solutions to movement problems are generated at as low a level in the hierarchy as possible; and 4) solutions are based on the invocation of pre-determined and rather abstract "ball-park" solutions that are refined using feedback and feedforward. [3, 13, 16]

Hierarchical decomposition into autonomous substructures is a powerful organizing principle and is reflected in a large variety of natural and artificial systems [5]. Such an approach is often found in robotics, the projected Mars rover under development at JPL being a good example [33]. There is a large body of literature concerned with the construction of legged locomotion devices [22, 30], articulated manipulators and prosthetic devices [28, 31]. When articulated motion is to be modelled, decomposition of the motion system has often been along the lines of functional synergies.

We have chosen this approach as the basis for our complex motion animation system because it works, as witnessed by the success of natural and artificial movement systems, and because it offers the promise of a movement system that is general, relatively easy to understand and control, and extensible.

## 6. Representation Issues

Since the aim of this research is to provide new tools and techniques for the animator, the interaction of the animator and the movement system must be considered at the outset. System operands and operators, i.e. skeletons and motion commands, respectively, must be designed with the user interface in mind.

### 6.1. Skeleton Representation

We have chosen a linguistic representation for skeletons. A context-free grammar has been constructed for describing skeletons in terms of 1) degrees of freedom and movement constraints at each joint, and 2) a transformation hierarchy for articulated movement. Instructions for drawing the skeleton, i.e. descriptions of the length and shape of the limbs, have not been included in order to keep the description display independent. The language could later be extended, if desired, to include such information. The grammar is simple enough to allow compact representations of real skeletons, but general enough to allow arbitrary complexity. With the language, an animator is free to define detailed skeletons of mammals, birds, insects, or completely imaginary creatures.

The use of a language for skeleton representation in the form suggested here has several advantages. Users familiar with programming languages and animation languages should feel at home with these constructs. Also, this kind of description is reasonably close to a "natural language" description of a skeleton. And finally, a skeleton description can be altered and extended easily during an animation session with any text editor.

### 6.2. Movement Representation

Ultimately, the skeleton movement system is to be incorporated into an animation system, and thus its input will be a script of some sort. The representation of motion should be in a form readily incorporated into an animation script, and should be easy to use and understand. For these reasons, movement representation in our system will be in the form of constrained natural language for which the domain of understanding is defined at design time and is extensible by the animator.

We will call a movement description at the script level a "task". A task is a description of some specific motion or sequence of motions the figure is to perform, e.g., "Walk slowly to the door. Open the door." Since the number of such tasks is infinite, it is clear that the movement system must abstract from the task a lower level description of the skeletal movements it needs to perform. We will say that the system abstracts from a task description a list of "global functions". In the above example, the global functions would consist of walking, reaching, grasping, and so on. At the top level, then, the movement controller will input a task description and output a list of (possibly concurrent) global functions. We will have more to say about global functions in the next section.

## 7. Movement Control

In its overall conception, the movement control system is a three-tiered structure which models the synergic, hierarchical control structure of natural movement systems. At the top level is the figure controller and task interpreter which decomposes a task into a list of global functions. The second level consists of all the global functions the figure can perform, together with the associated motor programs for executing them. At the lowest level are the local motor programs, each of which is associated with a given synergy, i.e. a fixed set of joints in the skeleton data base (created by the parser from the skeleton description). It is the responsibility of the intermediate level motor programs to coordinate the invocation of the lower level LMPs, which alone have access to the skeleton data base.

## 7.1. Control Techniques

The problem of designing motor programs and LMPs is one of correctly analyzing some target movement into a set of primitive motions. For some kinds of motion, particularly gait cycles and arm movements, a great deal of analysis has already been carried out. [24, 29, 20]

### 7.1.1. Finite State Control

The finite state control techniques worked out by McGhee and his co-workers [23] provide a framework for constructing LMPs for gait. Motor control programs are modelled as finite state automata that drive the figure through a set of predetermined configurations. It is a more powerful technique than either the use of key-frames or key-positions since state changes depend on the current configuration of the skeleton as well as the passage of time. This means that knowledge about what the skeleton is doing during a given frame can be used to generate movement in the next frame. Moreover, as McGhee has noted, [24] finite state control offers computational advantages since motion solutions are not based on the solution of the equations of motion of a figure. Instead, the problem is one of maintaining the logical and geometric relationships observed in the figure.

For human gait, these relationships have been described in the work of Eberhart, Inman, and Saunders [10], completed shortly after the Second World War. They showed how the human gait cycle can be broken down into essential primitive motions consisting of motor programs for the pelvis, knees, and ankles. In addition, Hartrum [19] implemented at Ohio State University a gait simulation based on the work of Eberhart et al, providing a rich source of implementation data in addition to the guidance offered by an early attempt to simulate complex motion.

### 7.1.2. Control of Variable-Configuration Motion

Coordination of variable-configuration motions, such as reaching for an object, has been analyzed, by, among others, Vukobratovic [30], who developed a synergic control technique called "algorithmic control" for implementing particular tasks to be performed by an industrial manipulator; and by Saridis [28], who implemented a control system for cosmetic motion of a human prosthesis.

It has been observed [13, 21] that biological motor systems employ motor solutions which expend minimum energy. With this in mind, we note that a large part of everyday physical activity is spent executing rather habitual motions. That is, when one opens a door without thinking about it, the motion is likely to be very similar every time. This suggests that a very large class of motions can be implemented by predetermined approximations that can be made quite precise at run-time, using feedback from the figure as it moves.

## 7.2. Motor Programs

After the task interpreter has generated a list of global functions to perform, the correct global function controllers, i.e., motor programs, must be invoked. From the simple task "Carry the glass to the table", for example, the task interpreter must abstract the global function "walk". The task interpreter knows nothing about walking; it only knows which motor program to call, in this case the gait controller. Likewise, the gait controller does not know how to move the skeleton. Its sole function is to invoke LMPs in a correctly timed sequence, frame by frame, based on its read-only access to the skeletal data base, and messages it receives from the LMPs.

As an example, consider the motor program that would control a simple straightahead gait cycle. Such a gait controller would be invoked by the task interpreter with parameters specifying the rate and extent of motion. For gait there would be LMPs for swing phase and stance phase for each leg. There would be an LMP to control pelvic rotation and tilt. LMPs would also be needed to control movements of the shoulders, arms, and spine. The gait controller would invoke the LMPs with the proper parameters and in the proper sequence; it would continue the walk cycle until it detected that the target position had been achieved.

### 7.3. Local Motor Programs

An LMP is a program which can access a fixed list of joints and change the current rotation and translation values for those joints. It is the LMPs that do the work of driving the skeleton through whatever configurations make up a given movement; they function much as concurrent processes accessing a shared resource, i.e., the skeleton data base.

The LMP for swing phase, in the example above, would be invoked by the gait motor program with a parameter specifying the number of frames over which the stride was to extend. The swing phase LMP for the left leg, say, would control the left hip, the left knee, and the left ankle. When invoked, it would increment the hip and knee joints from some minimum values at toe-off to certain maximum values at full extension, and then decrement the hip angle until heel-strike. At heel-strike the LMP would signal completion to the gait controller, currently monitoring the other LMPs involved in walking, all executing concurrently.

Concurrency, of course, is actually simulated. For each frame, LMPs are invoked in turn. The skeleton data base is updated by the LMPs, the display controller traverses the transformation hierarchy, and the updated matrices are sent to the display routines. There may be cases where LMPs might conflict, as when, for example, a figure is to walk with arms holding some object instead of swinging at the sides. Such conflicts can be avoided by specifying for a given movement that certain LMPs are essential and cannot be preempted or interrupted, while other LMPs are nonessential and may be overridden, as is the case with arm swinging during gait.

The construction of LMPs is a process to be guided by intuition, observation, and empirical results. A measure of the complexity of human movement is the dearth of descriptive data available in the literature for use by simulation designers. There is no blueprint for constructing movement mechanisms extant in any of the relevant disciplines--robotics, biomechanics, or physiology. In this sense, the design and subsequent use of a computer graphics skeleton animation system could make a small contribution to our understanding of animal locomotion.

## 8. Environmental Working Sets

The use of feedback and goal-direction imply that the interactions between skeletons and their environment must be simulated. In an oft-quoted paper Clark [6] discusses hierarchical methods for structuring information in a graphical data base in order to speed up visibility and clipping algorithms. One of the many interesting ideas in his paper is that of a "graphical working set"--the set of objects in the data base that are in some sense "near" to the current field of view. The design of a complex goal-directed animation system dictates that skeletons possess some measure of autonomy. It would be desirable, say, for skeletons to perform automatic collision testing so that body parts do not appear to pass through each other or through objects in the environment. But if the data base is large, consisting of several thousand or hundreds of thousands of surface elements, it is clear that without imposing some structure on the data base such interactions would be unmanageable. It would simply be too costly to search through the entire surface element list to determine the immediate neighborhood of a particular skeleton as it moves through space.

One approach might be to try to extend Clark's notion of a graphical working set. Each figure in a scene would maintain its own environmental working set--the list of bounding boxes and surface elements in its neighborhood.

Turner Whitted [35] has built a display algorithm based on hierarchical methods for the purpose of more efficient hidden surface elimination. In his system a scene is represented as a tree. Each node in the tree represents a subspace of a parent node. The leaves of the tree represent indivisible subspaces--solid objects that can be displayed. Each subspace is represented as a rectangular parallelepiped. A ray tracing algorithm is used to display the scene. As a ray progresses through the scene space, it is checked against the bounding box at the top level. If it misses this box, the background is displayed for that pixel. If it penetrates, the search proceeds at the next level of the tree, until a leaf node is encountered and the surface there can be displayed, or the ray misses all the subspaces at a given level.

Whitted's system does not incorporate the idea of a working set, but it illustrates how a graphical database could be structured using a tree of bounding boxes. A major difficulty is the creation of the tree in the first place. Currently this is done by hand, with the assistance of various data base editing programs.

Now suppose we wish to animate a human figure moving through some environment. We must be able to detect, say, when the feet contact the ground during the gait cycle. Clearly, the gait motor program and associated LMPs should not have to search the entire surface element list to find those surface elements nearest the feet. The gait controller need only know about those surface elements close to the figure, i.e., its environmental working set. Likewise, if each moving part of the figure maintained its own working set, collision testing could be performed quickly, especially if each working set was organized as a hierarchy of bounding boxes.

If we extend this notion to an entire scene, every moving figure would maintain its own working set tree. Each tree would be structured to match the structure of its figure. For a human figure, the top node of the tree would point to the bounding box for the room or area the figure currently occupies. The next level might be bounding boxes for the environments of the upper and lower parts of the figure. The next level would contain bounding boxes for the arms and feet, and so on, through the tree. At each level there would be pointers to a list of surface elements that lie within or partly within the current bounding box. Each bounding box would be completely contained in the bounding box of the parent node.

As a figure moves, the figure controller would update the top level bounding box and the surface list pointers. Scene coherence could be taken advantage of here, since figures tend to move at measured pace through a scene. Thus bounding box updates could be performed relatively efficiently at the top level. Each parent node in the tree, including the top node, would test the updated surface element list against the transformed bounding boxes of its children so that the bounding boxes lower down could be updated.

In this way, the tree would be maintained and updated in a hierarchical fashion, and each figure, or part of a figure, need only deal with environmental data in its immediate vicinity.

## 9. Scenario

To illustrate the approach we are taking toward complex animation, let us consider the steps an animator might take in developing an animated sequence involving a human figure.

Just as many animation systems maintain canonical and primitive objects--spheres, cubes, and the like--the animator could call up a "standard" human skeleton. Using interactive software much like that implemented by Gillenson [14], and Parent [25] the animator could "customize" the standard skeleton, i.e., specify dimensions and other attributes of the various segments.

Once the physical properties of the skeleton have been defined, the animator could determine the variety and quality of movement intended for the figure at hand, starting from a standard repertoire of movements maintained by the system. This repertoire would represent the default movement capabilities of the standard skeleton, and would consist of the set of global functions, such as walking, reaching, and so on, that the skeleton could perform. For each standard global function there must already be constructed a motor program to execute it, and a set of local motor programs for the motor program to invoke.

In this movement-definition phase the animator will have the option of adjusting the parameterization of motor programs and LMPs. For example, the parameters affecting gait would be those identified by Eberhart et al [10] which are adequate to produce pathological as well as normal gait. Thus the animator could produce a figure with a limp if that was called for in the animation.

We see the movement-definition phase as an iterative process in which the animator makes a choice and sees the results displayed in short order. Simple sequences lasting only a few frames may be displayed completely in a matter of seconds; lengthier and more complex movements may have to be viewed using a real-time playback scheme. Nevertheless, we feel that it is essential that the process converge very quickly, that the animator achieve satisfactory results in a few iterations. One of the advantages of hierarchical, synergic control is the local control it offers. Changes made to motor programs and LMPs will not propogate unpredictably through the movement system.

One of our long-term goals is to provide the animator with the capability to add new motor programs and LMPs. This could be accomplished either through programming language constructs which are extensions of existing robotics languages such as AL [12], or through the use of digitizing devices which could be "led through the motions" of a new sequence in much the same way industrial robots are currently programmed for new tasks.

Once the movement repertoire of a figure has been arrived at, the animator would then be free to construct an animation script and include task descriptions for which global functions and motor programs have been defined. During the animation phase, the animator need only specify what the figure is to do. It is up to the movement simulator to execute and assemble the correct set of motor programs over a given series of frames.

Thus we see movement animation as a two-step process. In the first step the animator defines an instance of the standard skeleton and its movement repertoire with the assistance of interactive software and real-time playback. In the second step the movement simulator is invoked to execute the animation script using default motor programs, default motor programs with user-specified parameters, or new motor programs constructed by the user.

Although the animator is responsible, as always, for understanding how the figure is to move, such knowledge can be imparted to the movement simulator in the definition phase. The burden of generating movement in detail is then shifted to the movement simulator during the animation phase.

## 10. Summary

We have reviewed some of the developments in computer animation with an emphasis on systems for portraying complex motion. We discussed some of the difficulties of specifying and controlling computer models of articulated motion, and suggested that what seems to be called for is an approach that simulates natural, complex motion interactively.

The use of hierarchical, synergic control in natural and artificial movement systems was reviewed. We examined the implications of synergic control for movement animation systems that are general, extensible, and habitable. Next, some of the issues involved in skeleton and movement representation were discussed. We suggested that linguistic representations for both provided extensible, accessible interfaces for the animator.

Using control techniques from robotics a three-level control system will be used to update a skeleton data base. The three levels consist of a top-level task interpreter for decomposing task descriptions, a set of motor programs for controlling low-level local motor programs, and the local motor programs themselves which have sole access to the skeleton data base. An example was given to illustrate our approach to movement simulation. Finally, we suggested that an extension of the notion of a graphical working set to an environmental working set might facilitate the interaction between animated figures in a scene and their environment.

## References

1. Badler, N. and Smoliar, S., "Digital Representation of Human Movement," ACM Computing Surveys, Vol. 11 (1979).
2. Baecker, R., "Interactive Computer Mediated Animation," Ph.D. Thesis, MIT (1969).
3. Bernstein, N., The Coordination and Regulation of Movements, Pergamon Press, Oxford (1967).
4. Carlson, W., Marshall, B., and Wilson, R., "Procedure Models for Generating Three Dimensional Terrain," Computer Graphics, Vol. 14 (2) SIGGRAPH-ACM, (July 1980).
5. Chandrasekeran, B., "Natural and Social System Metaphors for Distributed Problem Solving," IEEE Trans. on Systems, Man, and Cybernetics, (To Appear).
6. Clark, James H., "Hierarchical Geometric Models for Visible Surface Algorithms," CACM Vol. 19 (10), pp. 547-554 (October 1976).
7. Crow, F. C., "A Comparison of Antialiasing Techniques," IEEE Computer Graphics and Applications, (January 1981).
8. Csuri, Charles, Hummingbird, The Ohio State University (1967). Film
9. Csuri, Charles, Real Time, The Ohio State University (1970). Film
10. Eberhart, H. D., Inman, V. T., and Saunders, J. B., "The Major Determinants in Normal and Pathological Gait," Journal of Bone and Joint Surgery, Vol. 35A (3), (July 1953).
11. Evarts, E. V., "Brain Mechanisms of Movement," Scientific American, (September 1979).
12. Finkel, R., Taylor, R., Bolles, R., Paul, R., and Feldman, J., "An Overview of AL, A Programming System for Automation," Proc. Fourth International Joint Conf. on Artificial Intelligence, (September 1975).

13. Gelfand, I. M., Gurfinkel, V. S., Tsetlin, M. L., and Shik, M. L., *Models of the Structural-Functional Organization of Certain Biological Systems*, MIT Press, Cambridge (1971).

14. Gillenson, M. L., "The Interactive Generation of Facial Images on a CRT Using a Heuristic Strategy," Ph.D. Dissertation, The Ohio State University (March 1974).

15. Gracer, F. and Blasger, M., "KARMA, A System for Storyboard Animation," *Proc. Ninth Annual UAIDE Meeting*, (1970).

16. Greene, Peter H., "Problems of Organization of Motor Systems," *Progress in Theoretical Biology*, Vol. 2, Academic Press, (1972).

17. Grillner, S., "Locomotion in Vertebrates: Central Mechanisms and Reflex Interaction," *Physiological Reviews*, Vol. 55, (2) (April 1975).

18. Gurfinkel, V. S. and Fomin, S. V., "Biomechanical Principles of Constructing Artificial Walking Systems," *On Theory and Practice of Robots and Manipulators*, Vol. 1, Springer-Verlag, (1974).

19. Hartrum, T. C., "Computer Implementation of a Parametric Model for Biped Locomotion Kinematics," Ph.D. Thesis, The Ohio State University (June 1973).

20. Herman, R., Wirta, R., Bampton, S., and Finley, F. R., "Human Solutions for Locomotion: Single Limb Analysis," in *Neural Control of Locomotion*, Plenum Press, New York (1976).

21. Inman, V. T., "Human Locomotion," *Canadian Medical Association Journal*, Vol. 94, pp. 1047-1054 (May 14, 1966).

22. Kato, I. and Ohteru, S., "Information-Power Machine with Senses and Limbs," *On Theory and Practice of Robots and Manipulators*, Vol. 1, Springer-Verlag, (1974).

23. McGhee, R. B. and Tomovic, R., "A Finite State Approach to the Synthesis of Bioengineering Control Systems," *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-7 (2) (June 1966).

24. McGhee, R. B., "Robot Locomotion," in *Neural Control of Locomotion*, ed. R. Herman, Plenum Press, New York (1976).

25. Parent, R., "A System for Sculpting Three-Dimensional Data for Computer Graphics," Ph.D. Thesis, The Ohio State University (1977).

26. Pearson, K., "The Control of Walking," *Scientific American*, (December 1976).

27. Reeves, W., "Quantitative Representation of Complex Dynamic Shapes for Motion Analysis," Ph.D. Thesis, University of Toronto (1980).

28. Saridis, G. N. and Stephanou, H.E., "Hierarchical Intelligent Control of a Prosthetic Arm," Tech. Report TR-EE 76-8, Purdue University (July 1976).

29. Vukobratovic, M. and Juricic, D., "Contribution to the Synthesis of Biped Gait," *IEEE Transactions on Bio-Medical Engineering*, Vol. BME-16 (1), pp. 1-6 (January 1969).

30. Vukobratovic, M., Stokic, D., and Hristic, D., "New Control Concept of Anthropomorphic Manipulators," *Mechanism and Machine Theory*, Vol. 12, pp. 515-530 (1977).

31. Vukobratovic, M., Hristic, D., and Dzigurski, O., "Industrial Anthropomorphic Manipulator for High Positioning Accuracy Requirements," *Proc. 8th International Symposium on Industrial Robots*, (1978).

32. Wein, M. and Burtnyk, N., "A Computer Animation System for the Animator," *Proc. Tenth Annual UAIDE Meeting*, (1971).

33. Weinstein, M., "Structured Robotics," *Proc. Fourth International Joint Conf. on Artificial Intelligence*, (September 1975).

34. Wessler, B., "Computer-Assisted Visual Communication," Ph.D. Thesis, University of Utah (1973).

35. Whitted, Turner and Rubin, Steve, "A 3-Dimensional Representation for Fast Rendering of Complex Scense," *Computer Graphics*, Vol. 14 (2), pp. 110-116 SIGGRAPH-ACM, (1980).