

Digital Representations of Human Movement

NORMAN I. BADLER AND STEPHEN W. SMOLIAR

*Department of Computer and Information Science, The Moore School of Electrical Engineering,
University of Pennsylvania, Philadelphia, Pennsylvania 19104*

There are many different approaches to the representation, within a digital computer, of information describing the movement of the human body. The general issue of movement representation is approached from two points of view: notation systems designed for recording movement and animation systems designed for the display of movement. The interpretation of one particular notation system, Labanotation, is examined to extract a set of "primitive movement concepts" which can be used to animate a realistic human body on a graphics display. The body is represented computationally as a network of special-purpose processors—one processor situated at each joint of the body—each with an instruction set designed around the movement concepts derived from Labanotation. Movement is achieved by simulating the behavior of these processors as they interpret their respective programs.

Keywords and Phrases: Human movement, parallel processing, notation systems, human body model, animation, computer graphics, simulation.

CR Categories: 3.42, 3.49, 3.65, 8.1, 8.2.

INTRODUCTION

There are many approaches to the problem of how to represent, within a digital computer, information concerning and related to the movement of the human body. This information has been used, in one form or another, by a wide variety of scientific disciplines. As of 1972, an annotated bibliography of body movement research, prepared by Martha Davis [DAVI72], contained almost a thousand entries. One of the major difficulties which confronts any researcher who wishes to approach this vast literature source, however, is an almost total lack of agreement on how movement should be described. It is almost as if each research project started from scratch with an arbitrary set of movement characteristics to be observed. These characteristics might concern the body's positions or the

mobile aspects of how a motion is executed. They could concern the movement of the entire body, selective movement of specific body parts, or even such subtle gestures as eye contact [DAVI75].

Digital representations of human movement may be based on various modalities. Film or videotape could be used to record the movements of an individual or a group from one or more points of view. This material could then be digitized and made available for processing for any conceivable application. The difficulty is that this scenario involves an explosive amount of data, most of which would probably be ignored in any given investigation. Using techniques of pattern recognition and scene analysis, it may be possible to extract positions of a moving human body from the successive frames of a film and generate a symbolic representation of the movements

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1979 ACM 0010-4892/79/0300-0019 \$00.75

CONTENTS

INTRODUCTION

1 MOVEMENT NOTATIONS

- Labanotation
- Eshkol-Wachmann Notation
- Dynamic versus Positional Considerations

2 REPRESENTATIONS OF THE HUMAN BODY

- Stuck Figures
- Surface Models
- Volume Models

3 REPRESENTATION OF MOVEMENT

- Key Frames
- Movement Functions
- Goal-Directed Behavior
- Constraints
- Animation by Simulation
- Assessment

4 AN ARCHITECTURE FOR A HUMAN MOVEMENT SIMULATOR

- Primitive Movement Concepts
- The Human Movement Simulator
- An Example

SUMMARY

ACKNOWLEDGMENTS

REFERENCES

[BADL75]; but this has yet to be attempted on "live" data.

An alternative modality for movement description is natural language. There are excellent examples of natural language descriptions of classical ballet, including a comprehensive textbook of Russian ballet technique [VAGA69] and a remarkably thorough "choreographic script" of the ballet *Giselle*, written in English [BEAU69]. In addition, many of the nonverbal behavior studies by Ray Birdwhistell [BIRD70] incorporate extensive descriptions in English. The problem is now the reverse of that for digitized imagery: the description may be compact; but actual physical reconstruction of the information requires a certain amount of knowledge and sophistication about human movement (for example, the conventions and idioms of classical ballet). Moreover, natural language descriptions are subject to ambiguity and unavoidable imprecision in specifying positions, dynamics, styles, and other aspects of movement.

What is needed is an approach to the representation of human movement which accepts the diversity of information sources

and yet provides conceptually tractable data and control structures for the expression of movement. Figure 1 shows a system with such a movement representation as the central structure. Various output processes transform this data structure into other modalities. For input, body movements observed on film or video may be recognized as instances of the "primitive" movement concepts in the representation. Natural language could also be used to describe human movements, these descriptions being translated into the same primitives. Conversely, the primitives may be translated into natural language to provide a readable commentary.

The present status of research on computer vision and natural language processing indicates that these areas do not provide adequate support to define a set of primitive movement concepts. Movement notation systems, designed to record human movement in symbolic form, are a more fruitful area of investigation, particularly as they tend to provide more expressive power than artificial languages for computer animation. On the basis of how the most successful movement notations have developed, the following desiderata are posed for a set of primitive movement concepts, particularly in their capacity to support an effective human animation system:

- 1) Both destinations (goals) and movements (changes) can be specified. A specific translation or rotation (for example) can be executed, or a destination or orientation achieved in a limited form of goal-directed behavior.
- 2) The body will be moved with an implicit respect for balance and support, independent of a conscious effort to maintain these states by the user.
- 3) Movement can be constrained by described relationships between body parts or other objects, such as physical contact, proximity, and surrounding.
- 4) The dynamics and phrasing of a movement should be separable from the spatial displacement of each body part and should be based on empirical evidence for human performance dynamics.
- 5) Collision detection between the body and itself, other bodies, and other ob-

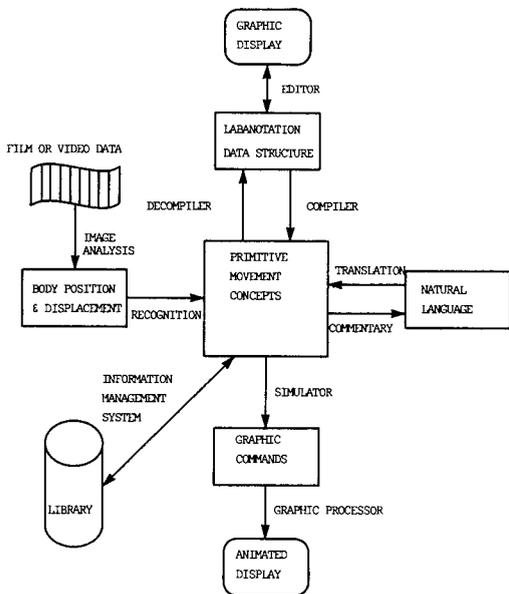


FIGURE 1. Comprehensive computer system for notating, modeling, analyzing, and describing human movement.

jects should be used for execution-time control and error monitoring.

- 6) Movement definitions should allow a flexible macro facility, including a repetition construct and parameter substitution.
- 7) The system should be tested on movement sequences noted for their scope and variety in exercising human capabilities. Thus, for example, it should be possible to reconstruct choreography recorded in a movement notation system.
- 8) The system should be capable of animating any human movement notation defined with sufficient rigor to admit reconstruction of the recorded movements. The notation must have well-defined semantics which can be translated into specifications for movement, against which the animation may be evaluated.

Section 1 discusses the features of several established movement notation systems, representation of the human body for computer displays is presented in Section 2, and methods for computer animation are reviewed in Section 3. Section 4 presents a specific system designed to accommodate the above desiderata.

1. MOVEMENT NOTATIONS

Labanotation

In approaching the problem of representing human movement within a digital computer, it is useful to observe how human beings communicate such information to each other. Most of the entries in the Davis bibliography [DAVI72] resort to what may be politely classified as "descriptive prose." However, two notations for the recording of human movement have been very systematically structured and provide valuable suggestions as to how such data might be organized within a computer. One of these, Labanotation (also known as Kinetography Laban), has been developed, revised, and extended since 1928 by Rudolf Laban and his colleagues; it is now maintained by a "standards organization," the International Council of Kinetography Laban [HUTC70]. The second was developed by Noa Eshkol and Abraham Wachmann [ESHK58] as a general-purpose movement notation and will be discussed in the following section.

Labanotation is based on an abstraction of the structure of the human body which is illustrated in Figure 2 [HUTC70]. The principal data elements of this abstraction are the individual joints and extremities of the body, with additional articulation of the torso region into "joints." (The illustration shows the Labanotation symbols used to represent these joints.) The essential task of Labanotation is to describe the position and trajectories of a set of points in space.

The position of each joint is specified with respect to a *cross of axes* which defines a rectangular coordinate system. This cross of axes may be oriented with respect to the room or the body in a variety of ways and is generally situated at a second joint of the body. For example, movement of the right lower arm may be achieved by positioning the right wrist with respect to a cross of axes situated at the right elbow. Alternatively, movement of the entire right arm is achieved by positioning the right wrist with respect to a cross of axes situated at the right shoulder. In describing any movement, the joint which is being positioned is called the *distal joint*, while the joint representing the limit of influence of the movement is called the *proximal joint*. The term

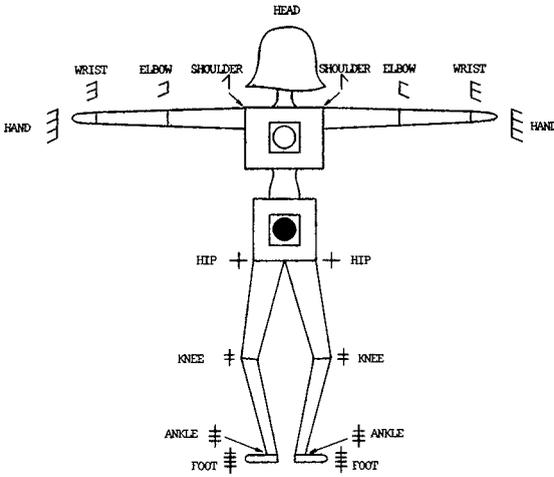


FIGURE 2 Abstraction of the human body into joints.

body part will be used to refer to that portion of the body which lies between a given proximal joint and a given distal joint [SMOL77b].

Movement may be expressed through five modes of description: 1) direction signs, 2) revolution signs, 3) facing signs, 4) contact signs, and 5) shape descriptions. Direction signs describe the translation of some joint, while revolution signs allow for the description of various forms of rotational movement, such as turning, twisting, and pivoting. Facing signs involve the establishment of an orientation of some point on the surface of a body part. Contact signs indicate surface contact of body parts with other body parts, other bodies, the floor, or other physical objects. Finally, shape descriptions are used to describe the tracing of a path or formation of a shape by some body part [SMOL77b].

Figure 3 [HACP70] is a representative sample of the notation, an excerpt from *Coppelia*; it is written on staves read bottom-to-top, the staves proceeding left-to-right. Below the staves are floor plans which indicate the patterns of movement on the stage. (These are the most common instances of shape descriptions.) The staves are divided into measures, each of which is divided into beats by small tick marks. The measures are numbered for correlation with the floor plans and with the accompanying music. Each staff is divided into columns, within which the movement symbols are written. The part of the body which per-

forms the movement is determined by the column in which the symbol appears.

Figure 4 [SMOL77a] summarizes the basic organization of the staff into columns and the principal structure of direction signs. Timing is determined by length, while direction in space is given by shape and shading. As one can observe in Figure 3, direction signs are the most predominant mode of description. The parallelograms in measures 6-8, 15, and 29-31 are instances of revolution signs; symbols in the outer columns of measure 5 are facing signs. (These particular facing signs indicate the orientation of the surface of the palms.) Finally, contacts are represented by the bow in measure 5 and by the small hooks which modify many of the symbols in the right-most staff.

Eshkol-Wachmann Notation

The structure of the human body is abstracted in a somewhat different manner for Eshkol-Wachmann notation, as illus-

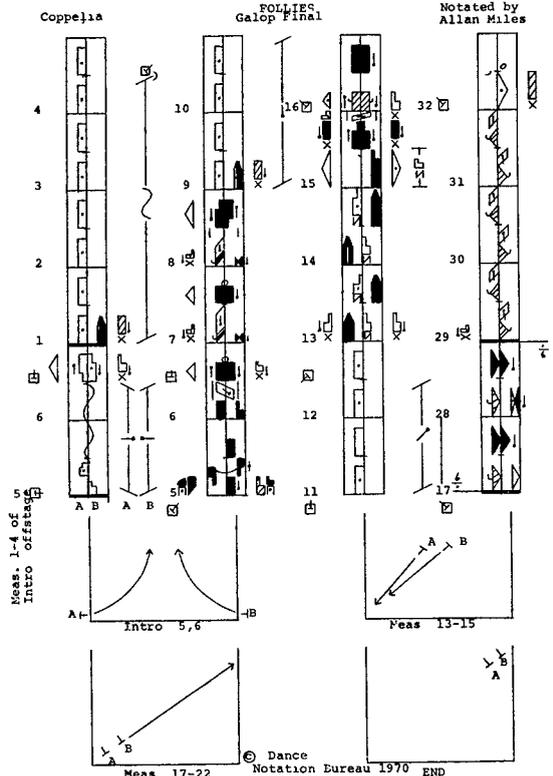


FIGURE 3. Example of Labanotation score excerpt from *Coppelia* [HACP70].

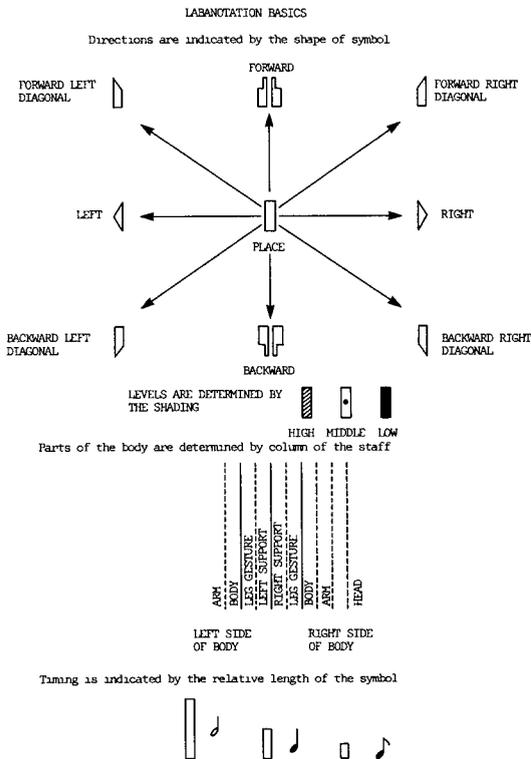


FIGURE 4. Labanotation basics [SMOL77a]

trated in Figure 5 [ESHK58]. While Labanotation views the body as a set of joints connected by limbs, Eshkol-Wachmann notation views it as a set of limbs connected at joints. Furthermore, each limb has associated with it a *longitudinal axis*. The orientation of each longitudinal axis is, again, determined with respect to a rectangular cross of axes. However, the situation is not as flexible as that of Labanotation: the cross of axes used to orient each longitudinal axis is uniquely fixed at that end of the axis which is closer to the center of the body. There is also a "law of 'Light' and 'Heavy' limbs" which states that when a given limb moves, it carries along those limbs which do not lie between it and the support of the entire body. For example, when the body is standing on both feet, movement of the right upper arm implies movement of the entire right arm as a rigid body, unless movements of the forearm and hand are explicitly notated.

The primary modes of description are similar to those which may be expressed by revolution signs in Labanotation. A move-

ment is a movement of a limb about its longitudinal axis. In a plane movement the longitudinal axis sweeps out a plane; thus it is moving at right angles to the axis of movement. In a curved movement the longitudinal axis sweeps out a curved surface, achieved by moving at an acute angle to the axis of movement. Finally, movement may be defined by specifying a position which a limb must assume.

Dynamic versus Positional Considerations

There is an alternative approach to movement description concerned with how the body concentrates its energies (effort qualities) and how the body uses energy to form itself in space (shape qualities) [DELL70]. This approach, known as Effort/Shape Analysis, has its own notation, which also originated with Laban. This notation is not sufficient for reconstruction of a movement pattern, but simply records in a compact form the presence of effort and shape qualities in an individual's movements.

Effort qualities may be described in terms of combinations of four parameters. The parameter of *tension flow* describes whether the movement is bound, i.e., consciously controlled, or relatively "free" (with respect to the build of the body). The *weight* parameter describes a quality of lightness or forcefulness. The *time* parameter indicates either a sustained or sudden



FIGURE 5 Body segment structure [ESHK58].

quality in the movement. Finally, the *space* parameter is concerned with the spatial focus of a movement—whether it is directed to a single focal point or indirectly divided among several foci. There are also four parameters for the description of shape qualities. Like effort, shape has a *flow* parameter, which characterizes movements as being toward or away from the body center. Then there are parameters for the three directional axes: *vertical* (rising or sinking), *sagittal* (advancing or retreating), and *horizontal* (widening or narrowing) [DELL70].

While shape qualities seem vague for a computational representation for movement, it is tempting to believe that effort qualities may translate into simple variations on the dynamics of a movement, such as those found in various computer animation systems [CATM72, MEZE71, SPEG75]. Unfortunately, this approach is probably insufficient: the problem is that Labanotation and Effort/Shape Analysis are founded on complementary models of the human body. Labanotation is concerned with *positional* information. It is therefore ideal for describing the orientation of the limbs of the human body at any given moment of time. We call its foundation *skeletal*: all descriptions may be formulated in terms of positions of the joints, coupled with an understanding of how these joints are interconnected.

Effort/Shape Analysis, on the other hand, is concerned with dynamic information. One might say that the issue here is not the “what” of movement but rather the “how.” While the presence of effort and shape qualities may be *observable* within transitions from one position to another, this is not an argument for their implementation strictly in terms of convenient dynamic movement patterns. In contrast to Labanotation, Effort/Shape Analysis is founded on a *muscular* model of the human body. While it may be possible to represent the muscular system in a data structure, current knowledge of muscular behavior is hardly as thorough as that of skeletal behavior. The structure of the skeleton, the degrees of freedom at individual joints, and the movement constraints at each joint may all be easily and accurately measured for any human subject. On the other hand,

electromyography is about the only measurement technique for muscular behavior; and the information it yields is not particularly accurate [GRIE76]. In fact, it even remains to be seen whether or not there is any meaningful correlation between the parameters of Effort/Shape Analysis and the actual behavior of muscle groups. Future research will have to verify such a correlation or determine if it is possible to modify the parameters of Effort/Shape Analysis for a more rigorous description of dynamic information.

2. REPRESENTATIONS OF THE HUMAN BODY

Computer animation can be broadly defined as the specification and display of the movements of objects. Animation of human movement therefore requires a specification of the body as an object for display and a suitable set of commands which operate on that specification to change its articulated form and its position in space. Since we desire data and control structures which will serve as a *model* of human movement, we cannot use traditional two-dimensional and 2½-dimensional (parallel overlapping planes) modeling techniques frequently employed by conventional and computer animators [BURT76]. Although a highly realistic rendering of a particular movement may be constructed using these techniques, the design process is not readily generalizable to the full range of body movements: the underlying model must be inherently three-dimensional in order to satisfy the desiderata stated in the Introduction. There are three general methods for modeling a complex curved three-dimensional object such as the human body. The limbs and joints may be abstracted as a stick figure, the curved surfaces may be explicitly represented, or a collection of volumes may be used to implicitly define the surface.

Stick Figures

Several systems for the display of human movement base their animations strictly on stick figures [BARE77, SAVA77, WITH70]; but this leads to two problems. First, the stick figure display (Figure 6 [WITH70]) is

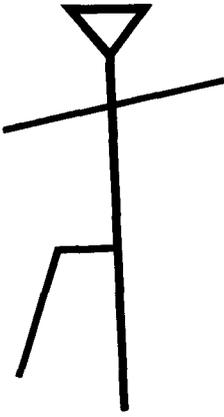


FIGURE 6. Stick figure model of body [WITH70].

enough unlike the usual appearance of a body to cause confusion in the perception of the animated movement. Depths are difficult to judge since body parts cannot occlude one another. This is particularly evident when the body as a whole turns about a vertical axis. With a simple stick figure it is very easy to confuse clockwise and counter-clockwise rotation. Second, significant classes of movements cannot be effectively portrayed, especially the rotatory movements of Eshkol-Wachmann notation (since only the longitudinal axis would be displayed), twists of certain body parts, and contacts between body surfaces.

The proper role of the stick figure is to model the network of body segments and joints which articulate the body. Like a skeleton, a stick figure describes the connectivity and flexibility of the body but often only suggests, rather than represents, exterior form. For such a representation a model of the body surface itself is needed.

Surface Models

The problems with stick figures are overcome by defining surface "skin" to surround the linear skeleton. The body surface may be modeled by partitioning it into planar or curved patches. Movements which displace the surface, such as rotations about a longitudinal axis, may now be visible; and suitable hidden surface removal algorithms [SUTH74] provide the proper occlusion and depth effects.

Surface points representing a planar decomposition of the body surface are used in

two models by Fetter [FETT64], one having 300 vertices, the other, 3000 (Figure 7). Although retaining the simplicity of the display primitives, this representation sacrifices the solid appearance of an actual body. While polygon models of the face [PARK72], hand [CATM72], and whole body [WESS73] have been used to obtain solid renderings, the display cost is quite high, since a large number of polygons are required. Furthermore, polygon models of a jointed shape may yield unnatural results when that shape is moved at a joint. For example, in Catmull's sequence of images of the hand (Figure 8 [CATM72]), joint movements deform the fingers by making them thinner as they bend. No provision is made to modify the planar vertices during the movement, and the appropriately interpolated transformations which might provide each vertex with a realistic movement would be nontrivial.

These difficulties are not resolved by using curved surface patches [ROGE76]. While the number of patches is drastically reduced because the surfaces are smoothly curved, the hidden surface removal process becomes more difficult [CATM75]. Curved patches are used in an experimental videotape of a walking man done at the New York Institute of Technology. During movement, the surfaces in the vicinity of an articulated joint may be deformed. This problem is potentially solvable, but it is not clear how many patches would be needed at each joint to model the surface appear-

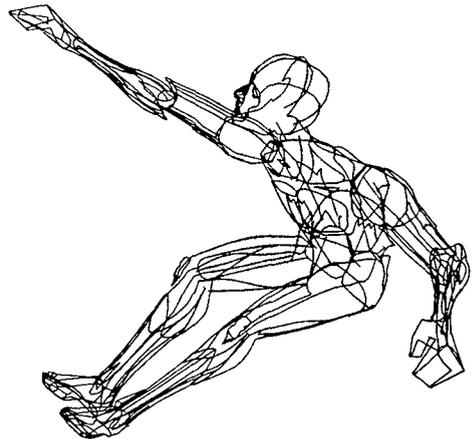


FIGURE 7. Surface point model [FETT64]

ance appropriately at different joint angles. Some deformations may even introduce singularities into the boundary curves.

Volume Models

The failures of surface modeling techniques are partially rectified by changing to a representation based on volumes. The body is decomposed into instances of one or more primitive volumes, such as cylinders [EVAN76, POTT75], ellipsoids [HERB74], or spheres [BADL78a]. A few cylinders or ellipsoids can capture the surface and longitudinal axis properties of many body parts,

although the resemblances are quite stylized: the forms are very smooth and symmetrical, even cartoon-like (Figure 9 [HERB74]). The primary difficulty with cylinders is that the planar end caps must be smoothed at the joints; Figure 10 [POTT75] shows how this can be done, but the best results are achieved only in front or side views. The cylinder end problem can be avoided by using ellipsoids, and hidden edges can be removed by a straightforward computation; but the result is not readily shaded. Hidden surface removal for shaded images of cylinders also tends to be extremely time-consuming [GOLD71].

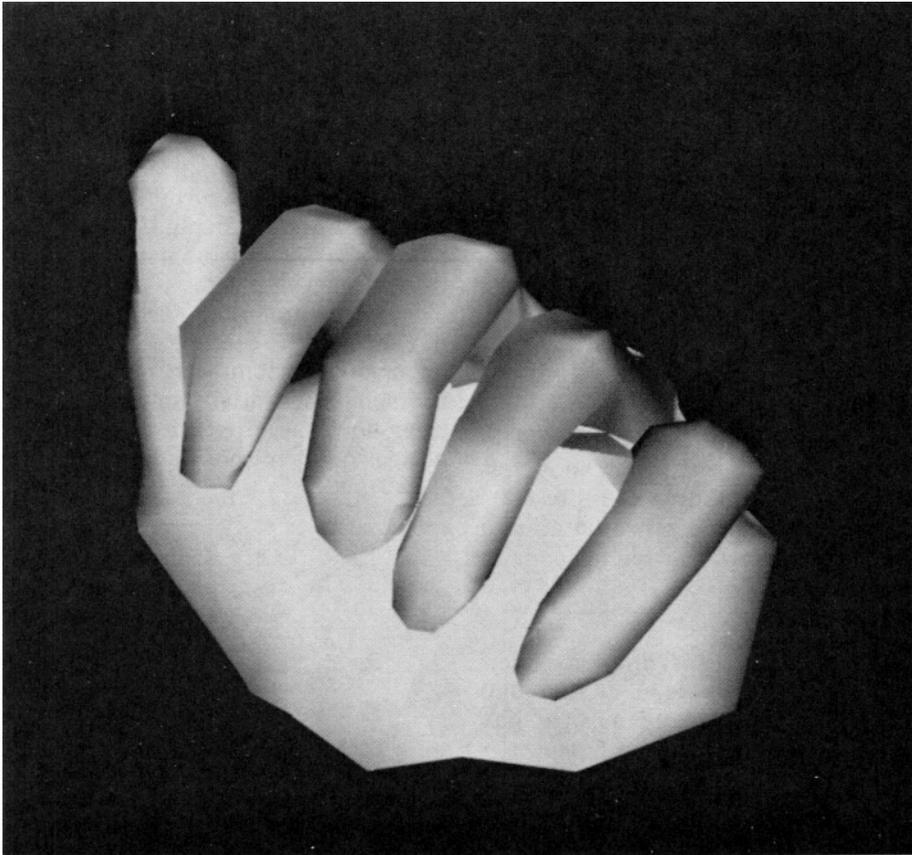


FIGURE 8. Hand modeled with planar polygons [CATM72]

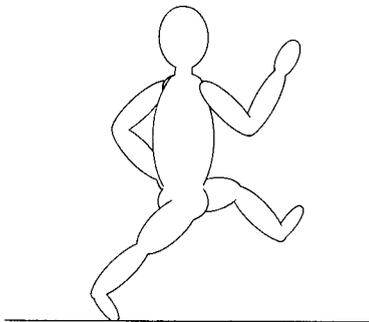


FIGURE 9. Body model using ellipsoids [HERB74].

A model using spherical primitives might seem counter-intuitive, but in fact such a model solves most of the problems variously encountered with other representations. About 300 spheres were used in the model of Figure 11 [BADL78a]. This representation may be used on either vector or raster displays, since each sphere projects into a circle or shaded disk. The natural overlapping of the spheres to approximate a curved surface makes hidden surface removal a simple variation on the z-buffer priority order technique [NEWE72]. Joint deformation problems disappear because the surface of a sphere at a joint is always defined, no matter what orientation the adjacent body parts assume. Arbitrarily shaped surface features, such as muscle masses, nose, and hands are easy to model since directionality is not an intrinsic property of a sphere (as in an ellipsoid or cylinder), only a collective property. Model generation, difficult for many object representations, is greatly simplified by a program which automatically produces a spherical decomposition of an object presented as a set of planar cross-sections [OROU77]. Finally, the model permits a simple test for the collision of body parts, since it is dependent upon finding intersections between spheres.

The principal objection to spheres is the "bumpy" texture which results in the edges of the image. While part of this problem arises from rastering effects (and Figure 11 is even anti-aliased to smooth the projection of each disk), most is caused by the separation of the representation spheres. A related problem is described by Clark [CLAR76] for polyhedral or curved patch

representations: the level of detail used in the description of an object should be compatible with the expected resolution on the actual display. Automatic interpolation techniques can be applied to a spherical decomposition, but this enhancement has yet to be implemented.

3. REPRESENTATION OF MOVEMENT

A graphic object may be moved in several different ways. Conventional animation relies heavily on two-dimensional techniques such as key frames and interpolation ("in-betweening"), while computer animation usually expresses position and velocity as functions of time. Simulations of "real" systems provide another source for animations; for example, commands to a robot manipulator are interpreted by a computational model of the manipulator itself. As a result, the graphic output is decoupled from the method used to express the activities of the system. In addition, movement commands often permit the specification of goals and constraints in a convenient manner.

The following sections summarize five somewhat interdependent approaches to movement representation: 1) key frames; 2) velocity and position as functions of time; 3) goal specification; 4) constraint specification; and 5) general simulation. Each of these approaches has been taken for different applications requiring movement representation. They will be assessed on the

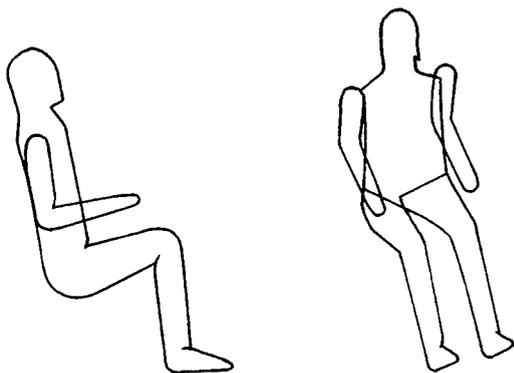


FIGURE 10 (a) Side view of cylinder model with smoothed end caps, (b) oblique view of cylinder model [POTT75].

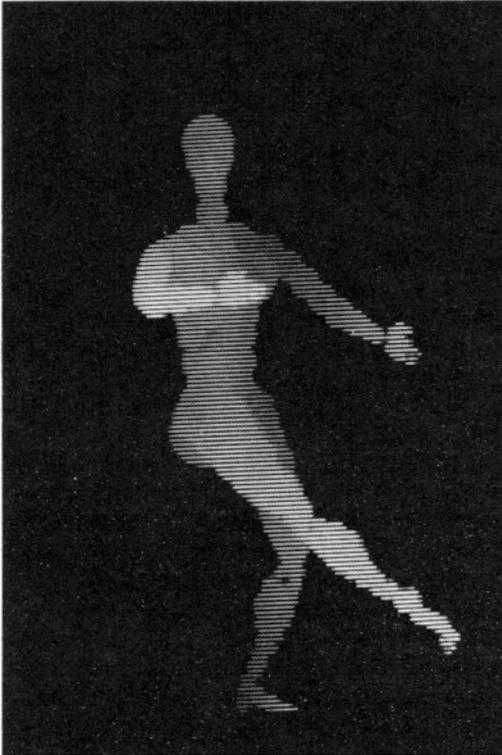


FIGURE 11 Human body modeled with spheres.

basis of the eight desiderata given in the Introduction.

Key Frames

Key frames are the primary tool of traditional animation. They are, essentially, those frames which provide the information most crucial to conveying the proper effects of movement. In animation studios key frames are drawn by the "master" animators, while the frames "in-between" the key frames may be relegated to the "journeymen" of the studio.

One of the earliest approaches to computer animation consisted in having the computer assume the role of these journeymen, their task being one of simply constructing interpolations [HACR77, MEZE71, WEIN77]. While some very effective animations have been achieved in this manner, they are fundamentally two-dimensional in origin and awkward to apply to three-dimensional objects. Because in-between frames are generally linearly interpolated, the movements may only approximate ac-

tual trajectories and also tend to deform the object. To reduce these effects, a two-dimensional skeleton technique [BURT76] can be used. While this technique improves the trajectory and helps preserve a mathematical or artistic form, the animator must choose his key frames subject to the constraints of a known interpolation algorithm. This tends to restrict his control of the actual flow of the movement, so interactive refinement of a specific sequence is essential.

Movement Functions

There are two approaches to defining movement by functions of time which may be interpreted on a frame-by-frame basis. The first involves specifying the path of an object point over time. This is the basis for the "p-curve" technique employed by Baecker [BAEC69] and extended to three dimensions by Csurí [CSUR75]. The second approach is based on geometric transformations, such as translation, rotation, and dilation (scaling), and possibly varying the velocity as a function of time to modify otherwise linear dynamics [CATM72, MEZE71, SPEG75].

Floor plans in Labanotation (Figure 3) demonstrate the primary strength of the path technique. Its principal weakness lies in the difficulty of describing any but global movements of an object. Describing the paths of subparts would require a variety of coordinate systems such as those discussed in the section, Labanotation. Coordination would be extremely difficult to specify strictly in terms of such path functions.

Eshkol-Wachmann notation, on the other hand, essentially takes the approach of geometric transformations. The problem here is one of intuition: it is very difficult for a human reader to grasp human movement strictly in terms of a composition of joint rotations. Thus, while such a representation may be very effective at a low level of implementation, it is not necessarily conducive to a general representation of movement concepts.

Goal-Directed Behavior

A paradigm for goal-directed behavior can be found in modeling and control systems

for robot "arm" manipulators. The programming languages AL [FINK75] and LAMA [LOZA76] are good examples of systems for the specification and control of a manipulator. The significance of these languages for an animation environment is the subjugation of the mechanical capabilities of the arm (or arms) to a specific task, usually part orientation and assembly. While trajectories and control signals are modeled within the AL interpreter, the user (ideally) sees none of this. The representation of manipulator actions is expressed with concurrent processes, movements to and from positions defined in arbitrary reference frames, and user-defined macros providing control sequences for semantically significant events (for example, grasp, release, or search). The user may also define monitors which are invoked whenever certain conditions become true. The implication of these flexible control structures is that a movement is described in terms of its goal or effect, rather than in terms of the mechanism structure.

This point of view is highly desirable for human movement because it offers the animator the freedom to ignore detailed specifications of how a movement may be carried out. The capability for exact joint movement is not sacrificed; rather the burden of excessive detail is lightened for the user who can now rely on built-in semantics and the structure of the body to achieve a goal. With AL, for example, the manipulator configuration is the responsibility of a set of "servos," one for each joint. Servos are created by the interpreter in response to compiled movement commands. Similarly, in the human movement simulator to be discussed in Section 4 [SMOL77b], each joint of the body is controlled by an independent processor in communication with a monitor.

There is a difference in the capabilities required for robot manipulation and human movement simulation, however. In AL the trajectory of the manipulator is constrained by providing arrival and departure vectors and an optional set of "via" points which guide the arm past known obstacles. The trajectory is then computed as a suitable polynomial curve [PAUL72]. In human movement none of this information may be available or else is dependent upon the

particular joint being moved, the current position and destination (but not always constrained to particular directions), and the magnitude of the displacement.

The danger with specifying fewer details of a human movement is that the trajectory defaults used by the goal-seeking system may not create the proper "effect" desired by the individual providing the specifications. The form or shape of the movement, its dynamics, and the manner in which it is articulated over several joints may be just as important as the goal. (In fact, key frames may be viewed as goal states filled in by two-dimensional, rather than three-dimensional, movements.) Movement descriptions must therefore be cognizant of the trajectory defaults, which may then be superseded with more precise instructions when necessary.

Both key frames and goal-directed movement description regard movement as a *process* implicitly defined as the transitions necessary to link together a specified sequence of states. The behavior of these transitions, however, is not always intuitively obvious; so it is not necessarily very accurate to refer to these techniques as *movement* representations. Alternatively, Labanotation offers an established set of semantics, including trajectories and methods of accomplishment, for most human movements. It is a system which describes movement *explicitly* and has a well-structured set of options to accommodate various refinements of a description.

Constraints

There is another type of movement description, related to goal specification and based on constraint satisfaction. For human movement such constraints usually express relationships between objects: contact, guide, support, and surround. In AL and LAMA, manipulator actions such as "grasp" describe constraints as well as a goal. Path functions essentially describe trajectory constraints.

Constraints also appear as physical limitations on the movement of an articulated object. The kinematics of linkages have been examined for manipulators [LOZA76, PAUL72] and for general two-dimensional linkages [SPEG75]. Spiegel [SPEG75] devel-

oped a language for expressing the attachment and support constraints of a mechanism and proposed various heuristics for distributing movement over several joints in a "natural" way. The instructions for a "walking" sequence are relatively easy to express as joint rotations, velocity functions, and constraints; but unfortunately the techniques are only two-dimensional. Although the walking movement is awkward, it is not caused by just the lack of the third dimension, so much as by the failure to constrain the balance of the figure. Since Spiegel's mechanisms are massless, this would seem to inherently restrict the system's capabilities.

A movement description does not supply all constraints explicitly: each joint of the body has movement limitations, and two objects cannot occupy the same space at the same time. Joint stops are simple to model as rotation constraints at different joint orientations, but collision testing is more complicated. The AL system has no built-in provision for collision testing (it is assumed to be the user's responsibility); but there is no reason why a collision detector could not be added as a condition monitor. This is done in LAMA, where the objects, the work station, and the available space are modeled by sets of rectangular volumes. The arm itself is modeled by cylindrical and rectangular volumes. Collision detection is essentially a matter of (efficiently) discovering intersections between objects and arm.

Animation by Simulation

When simulation is the vehicle for effecting movements of an object, graphics are completely decoupled from the physical model. The behavior of the model, here a human body, is dictated by the structure of the body as an articulated three-dimensional object interacting with itself and its environment. Although programs exist for the simulation of human body movements influenced by outside forces (such as vehicle crash studies [ROBB72]), we are more concerned with behavior originated by forces within the body.

Movements of any three-dimensional model may be described by positioning

joints or body surfaces in space or in relation to one another. The latter characteristic is the essential difference between simulation and the other animation techniques: movements described as *relationships* may depend upon positions of the body achieved by other concurrent movements. The only way these interacting body parts may be related is by performing the movements and modifying the body position. Moreover, each body part has a different capability for movement: the semantics of a movement depends upon the particular parts being moved, whether or not they support weight, the intermediate joints which may be involved, the shape of body segments, and any limits to joint angles and segment twists.

Assessment

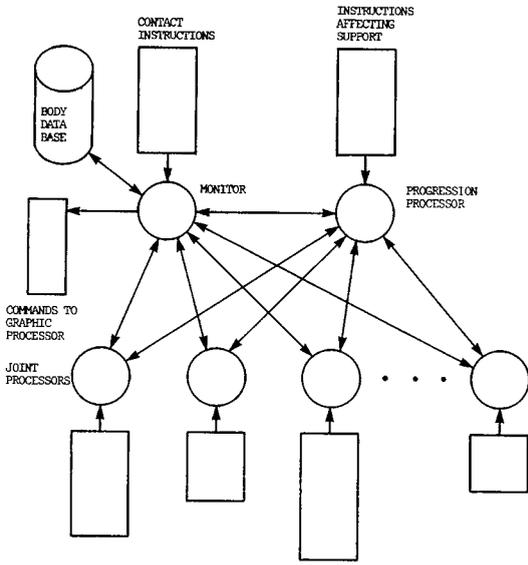
The above five approaches to movement representation may now be assessed in terms of the eight desiderata presented in the Introduction. This assessment is summarized in Table I. The results, as presented, favor animation by simulation because simulation systems model the physical effect of concurrent movements in a general environment independently of graphical representation. The next section describes an architecture for a human movement simulator which has been proposed to satisfy these eight desiderata, particularly the criteria of testability and reconstructability.

4. AN ARCHITECTURE FOR A HUMAN MOVEMENT SIMULATOR

The Labanotation abstraction of the structure of the human body, described in the section, Labanotation, suggests a computational model for simulating human motion. We model the human body as a network of special-purpose processors—one processor situated at each joint of the body—each with an instruction set designed around a set of "primitive movement concepts." Thus, Figure 2 may be interpreted as a "first approximation" of an assignment of processors to body joints. (According to this terminology, the extremities of the body will also be classified as "joints,"

TABLE I. CAPABILITIES OF VARIOUS ANIMATION TECHNIQUES

Technique	Destinations and changes	Balance and support	Relationship of body to environment	Separate dynamics and movements	Collision detection and control	Macro definition	Reconstruction	Evaluation
key frames	generally	no	no	no	no	no	artistic results achieved	no
paths	limited	no	no	no	possible	limited	stylized animations	no
velocity and position functions	generally	no	no	yes	possible	yes	limited animations	possible
goals (AL [FINK75])	yes	possible but untried	yes	possible	possible	yes	untried	possible
goals (LAMA [LOZA76])	yes	possible but untried	yes	possible	yes	yes	untried, but limited in degrees of freedom	possible
constraints	yes	support only	contacts	possible using velocity functions	possible	yes	walking; arm gestures	possible
simulation	yes	yes	yes	possible	yes	possible	to be tried	yes



INDIVIDUAL STREAMS OF DIRECTION, REVOLUTION, FACING, AND SHAPE INSTRUCTIONS

FIGURE 12. Structure of human movement simulator

even though they do not articulate two limbs.) If necessary, this assignment may be further refined for greater detail; but for the purposes of this discussion, the detail in Figure 2 is sufficient.

The general structure of the interpreter is illustrated in Figure 12. It is important to realize that Figure 2 should not be confused with the communication structure among the joint processors. Rather, it represents the basic data structure of the human body maintained by a *monitor*. This monitor is also responsible for all information exchanged among the joint processors, as well as the synchronization of their real-time processing.

In addition, the monitor assumes primary responsibility for the interpretation of contact signs (one of Labanotation's modes of movement description), since it maintains a global view of all executing processes. This capability also obliges the monitor to detect and interpret contacts which were not specifically requested, in particular, collisions between body parts or other objects. Such objects may be represented by planar polygons or sphere sets (Figure 13).

All joint processors share a common instruction set which is based on the remaining four modes of description cited in the section, Labanotation: 1) direction signs, 2)

revolution signs, 3) facing signs, and 4) shape descriptions. Each mode of description may be represented as an instruction for a joint processor. In the sequel we shall summarize the information which each of these instructions must provide to the joint processor to enable its execution.

In addition to the five modes of description, there is a functional distinction between *gesture* and *support* movements. The latter differ from the former in that they are based on movement of the body's center of gravity. (The center of gravity will generally also be slightly displaced as a result of gestural movement; but, in a gesture, displacement of the center of gravity is an *effect*, while in a support movement its displacement is the *cause* of the movement.) Support movements are implemented by a *progression processor* (Figure 12), which is capable of dispatching commands to those joint processors involved in bearing the weight of the body. (These are generally the processors at the ankles.) Commands dispatched by the progression processor

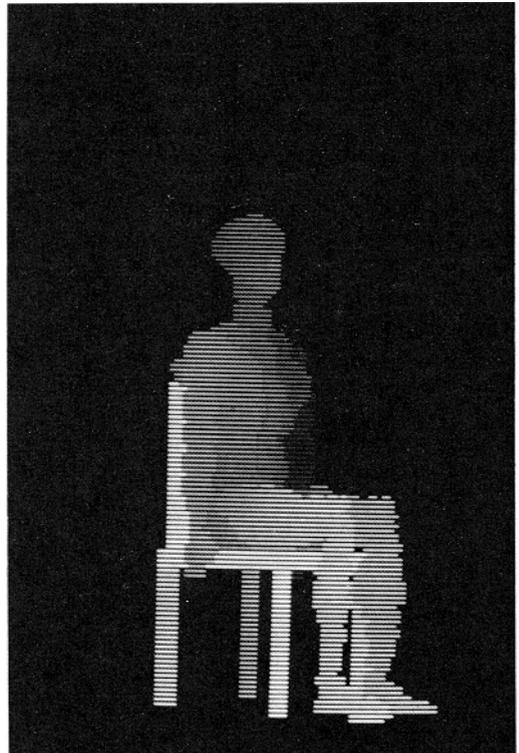


FIGURE 13 The body model may interact with its environment

take priority over instructions which implement gestures. Synthesized contact signs are also dispatched to the monitor to control the exact form and timing of the supporting movements.

The progression processor utilizes information regarding the center of gravity, the points of support, which joints affect the relationship between the points of support and the center of gravity, which supporting body parts are to move, and any overall shape which describes the movement path. This information is used to determine the new position of the center of gravity and to select the joint processors which will implement this positioning. Thus, "walking forward" is executed as a forward movement of the center of gravity to which supporting body parts (feet, for example) react to maintain proper contact with the floor. The movements of other joints (knees and hips, for example) are derived from the constraints supplied by the center of gravity path, supporting joint movement and contact timing [BADL78b].

Primitive Movement Concepts

Let us now consider what information each mode of description must provide to qualify as an instruction for one of the processors in Figure 12. A direction sign specifies the translation of a joint as either a *position description* or a *movement description*. The former describes the *orientation* of the distal joint of some body part with respect to a cross of axes. The latter describes a *path of motion* with respect to the initial position of the joint. The necessary components of a direction sign are duration, direction, designation of proximal joint, and the specification of either position or movement description. Optional components allow for modification of the path of motion, which may involve the movement of other joints, or specification of an alternative cross of axes.

Revolution signs specify movement about some axis. Consequently, the instruction must designate a duration, a proximal joint, an axis about which revolution occurs, the amount of revolution, and a descriptor to differentiate between twist and rotation. A *twist* is a revolution of a body

part where the proximal end does not move to the same extent as the distal end. (The lower arm twists.) A *rotation* indicates that all areas of the body part will turn uniformly. (This is seen when the body as a whole turns.) Finally, a modifier may be present to allow for alteration of the cross of axes used to determine the axis of revolution.

A facing sign requires three pieces of information. The first is the amount of time to establish the facing. The second provides a description of that part of the body surface whose orientation is specified, and the third specifies the orientation.

Direction signs, revolution signs, and facing signs are all interpreted in terms of a movement originated by a single joint. Contact signs, on the other hand, are executed by the monitor. Rather than providing durational information, a contact sign specifies the absolute time at which the contact occurs. (One of the functions of the monitor is to determine when the contact terminates.) It is also necessary to specify what is in contact; this may include body parts, the ground, or other structures (e.g., other persons, clothing, or objects). The types of movements specified by a contact sign are a *relationship*, which indicates an orientation, without actual physical contact; *nearness*, which also does not involve actual physical contact; *touch*; *bearing of weight*; and a *passive* approach to the relation. Modifiers include a specification of the contact taking place "in passing," the contact involving a surrounding movement, or the sustainment of contact.

Shape descriptions describe shapes of paths and shapes of configurations of body parts. (At a higher level they may also be used to describe shapes of groups of people.) They require a duration, designation of a proximal joint, a plot of points in three-dimensional space to describe the shape, and a designation of whether the shape indicates a position or movement description. A modifier may be present to alter the specification of the cross of axes.

The Human Movement Simulator

The actual interpretation of Labanotation is achieved by a two-stage process. In the

first stage, called the *compilation stage*, a set of disjoint programs for the individual processors illustrated in Figure 12 is prepared from Labanotation input. The interpretation of these programs is the function of the second stage, called the *simulation stage* [BADL78b]. During this stage, the monitor processor (Figure 12) is responsible for synthesizing the program which will be passed on to the graphic processor. All contact signs are collected together in a single program for the monitor. Instructions affecting support are sent to the progression processor, and all remaining instructions are placed in the instruction streams of their respective joint processors. Within the progression processor instruction stream, support instructions are further partitioned: each block accounting for the movement of a single supporting joint. This partitioning is handled by the compilation stage, since no instruction, in itself, indicates which joint processor it is meant to direct.

All timing information for a joint processor is provided by the duration fields of its instructions. However, block-structured parallelism enables the representation of concurrent execution of several instructions by a single processor [FEDA78]. Nevertheless, the *absence* of movement must be explicitly represented by a "null" instruction (analogous to a rest in music notation); and all substreams of a concurrency block must fill the same duration interval. All instructions express time in terms of a rational number of units. The time unit is related to the simulation process by defining a *simulation interval* to be the real time between successive "snapshots" of the human figure desired by the graphic processor. The simulation interval is represented as a nonzero rational number, where the only restriction is that no instruction may begin or end between simulator "snapshots." The simulation interval may be fixed by the user or may be computed by the monitor based on the earliest starting time of the upcoming set of instructions. (Each processor can supply this information to the monitor.) By permitting the interval to vary, the simulator can treat quiescent periods more efficiently.

The general control of the simulator involves the iterative execution of the following six steps for each simulation interval:

- 1) The monitor generates instructions to initiate contacts.
- 2) All current activities are represented by (joint processor, instruction) pairs; these pairs are assigned a priority ordering based on body structure.
- 3) The monitor allows the progression processor to implement any currently active support movements.
- 4) The monitor allows the implementation of each (joint processor, instruction) pair according to the priority order established in Step 2. Pairs with the same priority (and therefore the same joint processor) are executed concurrently by the joint processor.
- 5) The progression processor adjusts balance, if necessary.
- 6) The monitor calculates all final body positions and prepares the output for the graphic processor.

For each simulation interval, the monitor must first establish how contact instructions may be executed. Since contacts have no explicit duration prior to achievement, the monitor must utilize suitable existing instructions or synthesize new ones for appropriate joint processors. The monitor must next organize the actual execution of the other processes so that their sequential execution will in fact appear logically parallel. A priority ordering is computed by the monitor to insure an overall determinism to joint movements and to facilitate the manipulation of joint location information stored in the body database. Since a joint processor may be executing several instructions in a conceptually parallel fashion, a priority is computed for each active instruction of each processor. Should these priorities be the same, the processor itself establishes the order of execution within the set. The monitor now transfers control to the progression processor, signalling that modifications to the body database may be performed. After support movements, the joint processors execute their instructions based on the order scheduled by the monitor. When all the joint processors have com-

pleted their current instructions for this cycle, the monitor again calls upon the progression processor to balance the body, if necessary.

At the completion of all processing for a simulation interval, the monitor outputs a single stream of commands to a graphic processor which constructs and displays an image of a human figure. Over a sequence of simulation cycles an animation is produced. If desired, the monitor also generates a textual report of the model's position, contacts, and collisions.

An Example

As an example of the simulation process, consider the interpretation of the Labanotation segment illustrated in Figure 14. It describes one cycle of a normal forward walk: first the left foot steps forward, then the right foot follows. Because the direction signs for support actually describe the transference of weight, the feet do not move with respect to the floor during a forward direction sign: rather, the center of gravity moves forward [HUTC70]. During this forward movement of the center of gravity, the right arm first moves so as to point straight down from the shoulder, then

moves to a position slightly forward of the body; the left arm first moves slightly forward of the body then moves to the straight down position. (The final arm positions are shown in Figure 15.)

The Labanotation segment is compiled into three instruction streams: two direction signs for the left wrist, two more for the right wrist, and two for the progression processor:

left wrist processor:

- 1) (duration 1; to (place, low, forward); position)
- 2) (duration 1; to (place, low, place); position)

right wrist processor:

- 1) (duration 1; to (place, low, place); position)
- 2) (duration 1; to (place, low, forward); position)

progression processor:

- 1) (left ankle; at 0; duration 1; to (place, middle, forward); movement)
- 2) (right ankle; at 1; duration 1; to (place, middle, forward); movement)

(The notation is simplified for purposes of this discussion and does not strictly correspond to the actual instruction format.) The terminology for direction is taken from

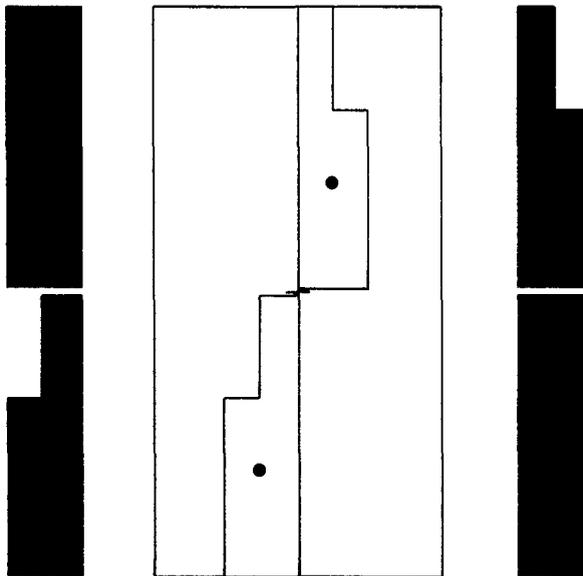


FIGURE 14 Labanotation fragment illustrating part of a walk

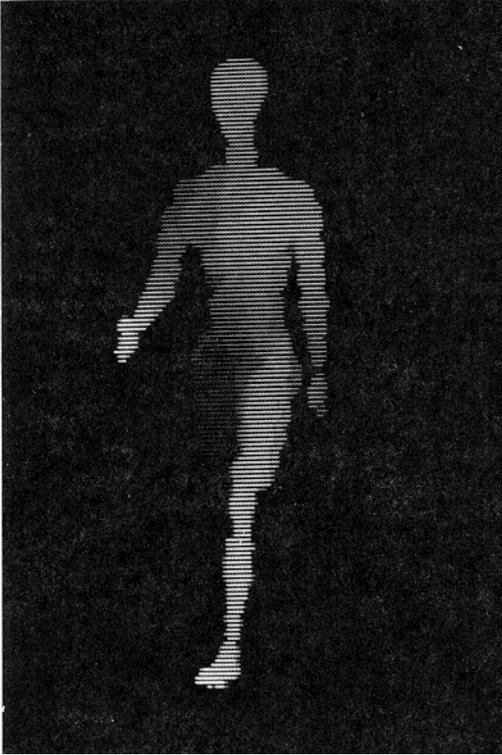


FIGURE 15 Body position at end of Labanotation segment of Figure 14

Figure 4; otherwise, the instructions are straightforward. For example, the first direction sign to the left wrist describes the position "pointing down and slightly forward of the body," which is to be achieved in one time unit. The other instructions are similar, except that the progression processor is explicitly informed of the starting times, movements, and identities of the supporting body parts.

Movements of the two arms are easily achieved by displacements of the wrists, since there are no other instructions which affect any other joints in either arm. These (joint processor, instruction) pairs, therefore, receive a low priority and are executed after support movements in each cycle. The two instructions to the progression processor each describe a forward "movement" (the section, Primitive Movement Concepts); the center of gravity is moved forward from its present location by a fixed amount, depending on the step length. Because a support movement has a "preparation phase" [HUTC70], execution of the

current instruction for a supporting joint depends on the instruction which follows. The progression processor must always look beyond those instructions it is currently executing to establish the proper context. In this example, the left heel strike actually occurs at the start of the segment; therefore, the position at this time will appear to be in the midst of the walk. When the final instructions to the progression processor are interpreted, it notes that there are no further instructions for the left ankle and therefore leaves the body balanced by bringing the center of gravity over the contact area of the right foot.

To achieve the appropriate leg movements implied by the support instructions, the progression processor generates contact instructions which are dispatched to the monitor. These contacts define the timing of the foot movements and—together with the step length, the movement of the center of gravity, and the geometry of the supporting surface—implicitly define the joint angles at the ankles, knees, and hips. For example, to prepare for the step onto the right foot, the progression processor issues two contact instructions to the monitor: one to break the right foot contact with the floor at time 0.5, and the second to achieve a right heel contact with the floor at time 1.0 (the beginning of the first progression processor instruction to the right ankle).

In this example a complex movement has been specified by a few instructions. However, much of this complexity is accounted for by default conditions which may be overridden by additional detail in the instructions. By choosing a very small simulation interval, smooth animations can be produced. While this would divide each movement into many intermediate positions, there would be no additional overhead in the number of instructions actually sent to each processor.

SUMMARY

In seeking a digital representation of human movement, established movement notation systems provide a wealth of well-structured information. One such system, Labanotation, has led to the design of a "machine language" based on a set of primitive movement concepts. Programs in this

language can be interpreted by a simulator to produce an animated display of human movement. A computer may be provided with such programs through a straightforward compilation of the symbols of Labanotation, through data provided by visual input, or through natural language. Thus, this "machine language" provides a highly flexible approach to movement representation.

Components of the movement representation system illustrated in Figure 1 are being constructed by the authors, based on the Labanotation abstractions described above. Several processes are operative, including:

- 1) A graphic editing system for Labanotation [BROW78, HIRS77];
- 2) A compiler from Labanotation to primitive movement concepts [FEDA78]; and
- 3) A display system for the human figure [BADL78a].

The display program was used to create Figures 11, 13, and 15, given joint angle inputs.

Scenery items, such as the chair in Figure 13, are constructed from lists of planar polygons.

The simulator is designed [BADL78b] and implemented with the exception of the detailed semantics of each joint. While the simulator itself is not expected to produce graphic commands at a real-time rate, these commands will be stored in a file and interpreted in "batches" by the display program. We expect that this process will be fast enough to animate the body model (drawn with circles to represent each sphere) on a graphics configuration consisting of a PDP-11/60 computer and a Vector General 3404 refresh display. Sequential snapshots may be produced on our Ramtek GX-100B color video display to obtain permanent video or film records of the solid figure in motion.

ACKNOWLEDGMENTS

The work reported on in this paper, which constitutes our own contribution to research into the digital representation of human movement, was funded by a grant from the National Science Foundation, Grant No. MCS-76-19464. We are particularly indebted to Lynne Weber, a professional dance notator, who has worked on this project with an enthusiasm which rivals that of the principal investigators. The students

who have contributed to this project are too numerous to mention, but we do wish to thank Hasida Toltzis, Joseph O'Rourke, Wayne Tracton, Deane Yang, and John Fedak, whose contributions were particularly influential in the writing and illustration of this paper. Finally, we wish to thank our referees for several extremely valuable suggestions and contributions

REFERENCES

- BADL75 BADLER, N. *Temporal scene analysis. conceptual descriptions of object movements*, Tech Rep No 80, Dept. Computer Science, Univ Toronto, Feb. 1975
- BADL78a BADLER, N., O'ROURKE, J., AND TOLTZIS, H. *A human body modelling system for motion studies*, Movement Project Rep No. 13, Dept Computer and Information Science, Univ. Pennsylvania, July 1978
- BADL78b BADLER, N., SMOLIAR, S. W., O'ROURKE, J., AND WEBER, L. *The simulation of human movement by computer*, Movement Project Rep. No 14, Dept. Computer and Information Science, Univ Pennsylvania, July 1978.
- BAEC69 BAECKER, R. "Picture-driven animation," in *Proc AFIPS 1969 Spring Jt. Computer Conf*, Vol. 34, AFIPS Press, Montvale, N.J., pp. 273-288
- BARE77 BARENHOLZ, J., WOLOFSKY, Z., GANAPATHY, I., CALVERT, T W, AND O'HARA, P "Computer interpretation of dance notation," in *Computing in the humanities*, S. Lusignan and J. S. North (Eds.), The University of Waterloo Press, Waterloo, Ont., Canada, 1977, pp. 235-240
- BEAU69 BEAUMONT, C *The ballet called Giselle*, Dance Horizons, New York, 1969
- BIRD70 BIRDWHISTELL, R. L. *Kinesics and context: essays on body motion communication*, University of Pennsylvania Press, Philadelphia, 1970.
- BROW78 BROWN, M, SMOLIAR, S W, AND WEBER, L "Preparing dance notation scores with a computer," *Computers and Graphics* 3 (1978), 1-7.
- BURT76 BURTNKY, N., AND WEIN, M. "Interactive skeleton techniques for enhancing motion dynamics in key frame animation," *Commun. ACM* 19, 10 (Oct. 1976), 564-569.
- CATM72 CATMULL, E. "A system for computer generated movies," in *Proc 1972 ACM Annual Conf*, Vol. 1, ACM, New York, pp 422-431.
- CATM75 CATMULL, E "Computer display of curved surfaces," in *Proc IEEE Conf. Computer Graphics, Pattern Recognition, and Data Structure*, 1975, IEEE, New York, pp 11-17
- CLAR76 CLARK, J "Hierarchical geometric models for visible surface algorithms," *Commun ACM* 19, 10 (Oct. 1976), 547-554.
- CSUR75 CSURI, C "Computer animation," *Computer Graphics* 9 (Spring 1975), 92-101
- DAVI72 DAVIS, M. *Understanding body movement. an annotated bibliography*, Arno Press, New York, 1972
- DAVI75 DAVIS, M *Towards understanding the intrinsic in body movement*, Arno Press, New York, 1975.
- DELL70 DELL, C *A primer for movement de-*

- scripton, Dance Notation Bureau, Inc., New York, 1970.
- ESHK58 ESHKOL, N., AND WACHMANN, A. *Movement notation*, Weidenfeld and Nicolson, London, 1958.
- EVAN76 EVANS, S M. *User's guide for the programs of Combiman*, Rep. No. AMRL-TR-76-117, University of Dayton Research Inst., Dayton, Ohio, 1976.
- FEDA78 FEDAK, J. *An initial design specification of a syntactic analyzer for Labanotation*, Movement Project Rep. No. 10, Dept. Computer and Information Science, Univ. Pennsylvania, Jan. 1978.
- FETT64 FETTER, W. A. *Computer graphics in communication*, McGraw-Hill, New York, 1964.
- FINK75 FINKEL, R., TAYLOR, R., BOLLES, R., PAUL, R., AND FELDMAN, J. "An overview of AL, a programming system for automation," in *Proc Fourth Int. Jt Conf. Artificial Intelligence*, Tsiblisi, USSR, 1975, pp. 758-765.
- GOLD71 GOLDSTEIN, R A., AND NAGEL, R "3-D visual simulation," *Simulation* 16 (Jan 1971), 25-30.
- GRIE76 GRIEVE, D. W. "Electromyography," in *Techniques for the analysis of human movement*, D W Grieve, et al., Princeton Book Company, Princeton, N.J., 1976, pp 109-149
- HACP70 HACKNEY, P, MANNO, S., AND TOPAZ, M. *Elementary reading studies*, Dance Notation Bureau, Inc., New York, 1970.
- HACR77 HACKATHORN, R. J. "Anima II a 3-D color animation system," *Computer Graphics* 11 (Summer 1977), 54-64.
- HERB74 HERBISON-EVANS, D. "Animated cartoons by computers using ellipsoids," in *Proc 6th Australian Computer Conf.*, 1974, pp. 811-823.
- HIRS77 HIRSCH, V. "The implementation of floor plans in the graphic editor for Labanotation," MSE Thesis, Dept Computer and Information Science, Univ. Pennsylvania, 1977
- HUTC70 HUTCHINSON, A. *Labanotation*, Theatre Arts Books, New York, 1970.
- LOZA76 LOZANO-PÉREZ, T *The design of a mechanical assembly system*, Rep AI-TR-397, Artificial Intelligence Laboratory, MIT, Cambridge, Mass., Dec. 1976
- MEZE71 MEZEI, L, AND ZIVIAN, A. "ARTA, an interactive animation system," in *Proc. IFIP Congress*, 1971, North Holland Pub., Amsterdam, pp. 429-434
- NEWE72 NEWELL, M. E., NEWELL, R G, AND SANCHA, T L "A solution to the hidden surface problem," in *Proc 1972 ACM Annual Conf.*, Vol. 1, ACM, New York, pp 443-450.
- OROU77 O'ROURKE, J. *Representation and display of three dimensional objects with spheres*, Movement Project Rep No. 7, Dept. Computer and Information Science, Univ Pennsylvania, August 1977.
- PARK72 PARKE, F. I. "Animation of faces," in *Proc. 1972 ACM Annual Conf.*, Vol. 1, ACM, New York, pp. 451-457.
- PAUL72 PAUL, R *Modelling, trajectory calculation and servoing of a computer controlled arm*, Memo No. 177, Artificial Intelligence Project, Stanford Univ., Stanford, Calif., Nov. 1972
- POTT75 POTTER, T E., AND WILLMERT, K D. "Three-dimensional human display model," *Computer Graphics* 9 (Spring 1975), 102-110.
- ROBB72 ROBBINS, D. H., BENNETT, R. O, JR, AND BOWMAN, B "User-oriented mathematical crash victim simulator," in *Proc 16th Stapp Car Crash Conf.*, 1972, pp 128-148
- ROGE76 ROGERS, D F, AND ADAMS, J. A *Mathematical elements for computer graphics*, McGraw-Hill, New York, 1976
- SAVA77 SAVAGE, G. J., AND OFFICER, J M. "Interactive computer graphic methods for choreography," presented at the *Third Int Conf Computing in the Humanities*, 1977
- SMOL77a SMOLIAR, S W, AND WEBER, L. "Using the computer for a semantic representation of Labanotation," in *Computing in the humanities*, S. Lusignan and J. S. North (Eds.), The University of Waterloo Press, Waterloo, Ont., Canada, 1977, pp. 253-261.
- SMOL77b SMOLIAR, S W, WEBER, L., AND BADLER, N. *An architecture for the simulation of human movement*, Movement Project Rep No 8, Dept. Computer and Information Science, Univ. Pennsylvania, Oct. 1977.
- SPEG75 SPEGEL, M. *Programming of mechanism motion*, Tech Rep No. CRL-43, Div Applied Science, New York Univ., New York, Nov 1975
- SUTH74 SUTHERLAND, I., SPROULL, R, AND SCHUMACKER, R "A characterization of ten hidden-surface algorithms," *Comput Surv.* 6, 1 (March 1974), 1-55.
- VAGA69 VAGANOVA, A *Basic principles of classical ballet Russian ballet technique*, A Chujoy (Transl.), Dover Publications, New York, 1969.
- WEIN77 WEIN, M "Computer animation," in *Encyclopedia of computer science and technology*, Vol. 5, Marcel Dekker, New York, 1977, pp. 397-436
- WESS73 WESSLER, B "Computer-assisted visual communication," PhD Dissertation, Dept. Computer Science, Univ. Utah, Salt Lake City, August 1973
- WITH70 WITHROW, C. *A dynamic model for computer-aided choreography*, UTEC-CSc-70-103, Dept Computer Science, Univ. Utah, Salt Lake City, June 1970.

RECEIVED JANUARY 18, 1978, FINAL REVISION ACCEPTED AUGUST 28, 1978