NUDES: A HUMAN MOVEMENT LANGUAGE
FOR COMPUTER ANIMATION

D. Herbison-Evans


BASSER DEPARTMENT OF COMPUTER SCIENCE
THE UNIVERSITY OF SYDNEY
N.S.W. 2006 AUSTRALIA
Summary

Seven different methods in current use for specifying human movement are considered: visual demonstration, natural language, specialised language, Benesh notation, Labanotation, numerical positions and muscle tension. The prospects of adapting each of these for computer input are discussed. The NUDES language (Numeric Utility Displaying Ellipsoid Solids) is described and related to this previous discussion.
NUDES : A HUMAN MOVEMENT LANGUAGE
FOR COMPUTER ANIMATION

Introduction

From time immemorial, humans have sought to represent pictorially the shape of humans and the movements of humans. With the advent of computer graphics, there have been a number of attempts to draw human figures with computer controlled drawing devices. The simple reproduction of outline and grey scale pictures by computer graphic devices allows various manipulations of the shape and representations of the figure of considerable artistic merit\(^1\), but do not allow easily for natural articulation of the figure. Attempts to represent all or part of human figures which can articulate have been made for ergonomic\(^2\), car crash simulation\(^3,4\), choreographic\(^15\) and facial animation studies\(^19\).

The present study was initiated by the choreographer Phillipa Cullen, who died so tragically in 1975. She indicated that the general problem of describing human movement in an economical, explicit, easily comprehensible, unambiguous fashion, especially as a linear string of symbols, was far from solved.
Existing Languages

As a starting point, let us consider the five methods used for the communication of human movement currently employed by choreographers:

(a) visual demonstration: the choreographer performs the required movements himself in the view of the recipient, either directly or via a film or a television recording device\(^{(6)}\).

(b) natural language description: this depends on there being names for the parts of the body, for the articulation joints, for the directions in space, and on there being a number of adjectives and adverbs which can describe position, speed and acceleration.

(c) specialised language descriptions: for the well formalised styles of dancing which have a limited vocabulary of movement there exists a series of formal terms describing the postures and movements of that style. A string of these then describe a dance in that style. This applies, for example, to square dancing, modern ballroom dancing\(^{(7)}\), and particularly to the classical ballet of the French and Russian schools\(^{(8)}\).

(d) Benesh notation\(^{(9)}\): this is a two dimensional graphic language used to describe human movement, originated by Joan and Rudolf Benesh in London in 1947. It is drawn on a stave which is divided into bars corresponding to the bars of the music or other quanta of time. The marks in one frame are basically an analog record of projections onto a vertical plane, of the movements of the tips of the limbs and of important joints during that time quantum. Formal signs are used to indicate motion in a forward or backward direction out of the plane of the drawing.
(e) Labanotation\(^{(10)}\): this is another two dimensional graphic language originally devised by Rudolf Laban in 1928. Different areas across the stave are reserved for motions of the different parts of the body: the legs, arms, torso and head. Motions for each part in its area are marked during that time period for which they occur along the stave: i.e., time is continuous and treated in an analog fashion. The motions symbols for any part of the body are restricted to a finite number and indicate direction and amount of movement: i.e., position is quantised. Another human movement language has been used, not for choreography but for computer animation of a human figure:

(f) Numerical\(^{(18)}\): this merely gives a sequence of vector coordinates for translation, and angles with associated axes for total and partial rotation of the figure for every picture frame in the animation sequence.

One further human movement language is widely used:

(g) Muscle Tension\(^{(21)}\): it is easy to conceive of an analog of the body's own language, in which the points of application of the more important body muscles, and the masses and moments of inertia of the parts of the body are defined. Movement can then be specified by setting appropriate tension levels in various muscles.

Such a simulation may have some important applications, but is not useful for the communication of a movement sequence as it overlooks kinesthetic feedback, which in the body is used to convert the muscle tension language into a position-demand language.

Prospects for Computerisation

Each of the seven types of language for the communication of human movement can in principle be used as the input to a computer system which has as
its output a graphic display of an animated human figure. Using the demonstration of movement (type A), systems have been made which record the positions of various parts of the body of someone demonstrating the motion as a continuous function of time. This is then digitised and reproduced\(^{(11,12)}\). This might be useful for building up a library of specific movement combinations and even complete dances, but is of limited value for synthesis of new movement combinations.

Some useful pointers to a human movement language can be gained from natural language (type B). The concepts of identifiers for parts of the body, for joints and for spatial directions are easily incorporated into a computer language. Position, speed and acceleration present more of a problem. A further characteristics of natural language is the preponderance of names for very complex motions: e.g., Blair's ten modes of bipedal locomotion\(^{(13)}\): walk, double bounce walk, run shuffle, sneak, jump, tiptoe, fast run, strut and skip. Most difficult are those which specify motion only up to the point of physical contact with an object or some other part of the body.

In the formal dance descriptions (type C) this is taken further, and each technical term may be considered to be a name for a complex motion. Implementation as a sort of library of all of the finite number of these for some dancing style may then enable dances described this way to be demonstrated. There are problems associated with the interfacing of the larger macros, which are left to "common sense" in the verbal description, but these do not seem insuperable. A more fundamental problem is that such types of language can only be used to describe and portray very specialised types of movement.

Benesh notation (type D) is very much more general, and is for instance being used both to describe the motion of physically handicapped people and for the description of remedial exercises for them\(^{(14)}\). There are problems in using it for computer input which arise from the continuous nature of the specification of position in the language, but an attempt is being made to computerise this language\(^{(16)}\).
Labanotation (type E) is also quite general. An attempt has been made to computerise it\(^{17}\) with limited success. The quantisation of position and continuous specification of time are more appropriate for computerisation, but the language seems to suffer from some confusion of position with action; e.g., the same symbol is used (although in a different part of the stave) for "put the left arm high" and for "take a high step with the left foot", (i.e. lift the left foot high, move it forward and put it down again).

The numerical language (type F) may be considered to be the "machine code" of human movement languages. It is the one into which any others have to be translated in order to implement them on a computer controlled display and is itself a tedious and error prone way in which to write anything of substance.

NUDES

This Numeric Utility Displaying Ellipsoid Solids is an assembler type version of the numerical language above (F). Identifiers are used, as in Natural Language (A) for the parts of the body and the joints, and preset keywords used for action commands.

A sample of the use of the languages is given in Appendix B.

The translation is currently implemented in ANSI Fortran IV together with a program which takes the numerical language output and draws the appropriate pictures either on a Calcomp plotter or the printer page\(^{18}\).

A sample printer page frame is shown in Figure 1.

Syntax

Each command is confined to a single card unless an asterisk is encountered in which case scanning then switches to the start of the next card. This also provides for comments (after an asterisk) and spacing lines (asterisk in first column). Information on any card after that required for the command is ignored and may also be used for comments.
All identifiers have to be declared before use. One or more spaces 
are used as separators between keywords, identifiers and numbers.

The syntax is incomplete as it still allows some sentences which are 
meaningless. The syntax is described more formally in Appendix A.

**Semantics**

The declaration of each joint includes the coordinates of the joint 
with respect to each of the two named body parts which articulate at that 
joint.

The actual implementation of the drawing routines assumed that the 
body parts are ellipsoids, so a declaration of each of these parts consists 
of the keyword ELLIPS followed by $x$, $y$ and $z$ direction semi-major axes of 
the ellipsoid part.

The parts of the body are grouped into figures, and subsequent action 
commands can refer either to all the figures in the scene or to a named 
single figure.

A COPY command generates a new figure with the same part and joint 
measurements as a nominated existing figure. The identifiers of the parts 
and joints of the new figure are those of the old figure with the first 
character replaced by the first character of the identifier of the new 
figure.

The basic effects of many of the action commands have been described 
elsewhere\(^\text{1}\), although a number of slight changes have been made.

DRAWAL and DRAWFG cause a single frame to be drawn of all or or one of 
the figures. VIEW causes a displacement in the notional position of screen 
onto which the perspective outlines of the figures are projected (assuming 
the eye to be at the origin). GROALL and GROFIG cause all dimensions of 
al or one of the figures to be multiplied by a scaling factor. GRNDAL and 
GRNDFG cause all or one of the figures to be moved vertically so that the 
lowest point just touches the ground ($y=0$).
MVALTO, MVALBY, MVFGTO and MVFGBY cause absolute or relative
translations of all or one of the figures by the amounts specified in x, y
and z directions.

RTALPT, RTFGPT, RTALEL and RTFGEL cause a rotation about a point or
about the centre of a specified ellipsoid, of all or of one of the figures by
the specified angle about the specified axis.

BENDBY and BENDTO cause articulation of the figure containing the joint
specified at that joint, holding the nominated ellipsoid stationary. The
bend can be specified as relative by using the BENDBY command or absolute
using BENDTO. For the BENDBY command, the rotation is specified by the
angle and axis fixed in the stationary ellipsoid. For the BENDTO command,
the azimuth and elevation are specified that are required of the moving
ellipsoid's x axis relative to the stationary ellipsoid at the joint.

SPEED allows the animated movie to be speeded up or slowed down. If
the numerical argument of the SPEED command is positive, then that many
extra frames are interpolated between existing frames and the picture is slowed
down. If the argument is negative, all frames but every n-th frame are deleted,
where -n is the argument.

This facility is of value in checking out an animated sequence before
committing all of it to a recording medium.

In order to cause a set of actions to be repeated (e.g., the steps
in walking) the LOOP and REPEAT commands are used. They bracket the actions
which are to be repeated, the number of repetitions being specified by the
integer following the REPEAT command.

It is also often desirable to be able to interpolate a single action
over a series of frames. The LINR and QUAD commands serve this purpose and
enable any of the other actions to be completed during the span of the specified
frame numbers. This is a generalised form of the two dimensional keyframe
animation system used by Foldes(20). The LINR command causes a linear inter-
polation over the specified frames: i.e., the action will appear to be done at constant velocity. The QUAD command causes a constant acceleration for half the time, then an equal constant deceleration for the other half, in such a way that the total motion is the commanded amount. Thus actions controlled by the QUAD command start and finish at rest.

An interesting conflict arises if the span of frames referred to by a LINR or QUAD command includes one or more LOOP and/or REPEAT commands. In this case, the binding of the spatial displacements and rotation angle increments cannot be done in a single pass at compilation time, and a translation to the numerical motion language (18) with constant displacements and angles can only be done in two passes.

The problem is exacerbated if the language is extended to contain conditional statements and subroutines, in which case the binding can only be done at execution. However, if arithmetic expressions are also included, this binding must be deferred till execution so the problem evaporates, at the expense of changing from a movement compiler to an interpreter. The present program is a compiler, so these extensions have not been implemented.

Future Extensions

As well as these extensions: arithmetic expressions, conditional transfers, as well as possibly procedures and arrays, and any of the other paraphenalia of high level languages, a number of non-trivial design problems need to be solved.

A major problem is the specification of rotations. The present numerical method is, in the author's experience, very error-prone, particularly in the specification of the sign of the required rotation angle. Possibly the natural language method of solving this should be used, perhaps specifying positive and negative rotations about each of the x, y and z axes as left down, left up, left, right, and down and up, respectively.
Another major problem is how to specify and to determine the complex motions required to bring one part of a figure into some position by moving other parts. This is exacerbated by the need not to allow the parts of the body to interpenetrate, and also the limited angular movement available to most joints. An extension of this problem is the general need for a command for motion until some part of the figure contacts some other part or some other figure.

It will be a little while yet before we can automatically translate and animate the contents of such classics as "Holstering his gun, he leapt onto his horse, tipped his hat and galloped off into the sunset".
APPENDIX A: 'NUDES' LANGUAGE SYNTAX

<program>::=<declarations>
    <actions>
        STOP
    </actions>
</declarations>
<declarations>::=<figure decln>|<ellipsoid decln>|<joint decln>|<copycmd>
<figure decln>::=FIGURE<figurename><ellipsoidlist>
<figurename>::=<name>
<name>::="up to six alphanumeric characters"
<ellipsoidlist>::=</ellipsoidname><ellipsoidlist>|</ellipsoidname>
</ellipsoidname>::=<name>

<ellipsoiddecln>::=</ellipsoidname><semiaxis><semiaxis><semiaxis>
<semiaxis>::=<number>
<number>::=<posnumber>-<posnumber>
<posnumber>::=<integer>|.<integer>|<integer>,<integer>
<integer>::=<digit> <integer>|<digit>

<digit>::=0|1|2|3|4|5|6|7|8|9

<jointdecln>::=JOINT<jointname><ellipsoidname><jointcoord><jointcoord><jointcoord>
<ellipsoidname>::=<name>
<jointcoord>::=<number>

<copycmd>::=COPY<figurename><figurename>

<actions>::=<action>|<action><actions>

<action>::=DRAWLAL|DRAWG<figurename>|
VIEW<number><number><number>|
GROALL<number>|
GROF<figurename><number>|
GRNDAL|
GRNDG<figurename>|
MVALT<number><number><number>
MVFHT<figurename><number><number><number>
MVALB<number><number><number>
MVFGB<figurename><number><number>
RTALP<angle><axis><number><number><number>
RTFGP<figurename><angle><axis><number><number><number>
RTAPE<ellipsoidname><angle><axis>
RTFEG<ellipsoidname><angle><axis>
BENDJ<jointname><ellipsoidname><angle><axis>
BENDT<jointname><ellipsoidname><angle><axis>
LOOP|
REPEAT<number>|
SPEED<number>!

-angle ::=<number>

<axis>::=1|2|3
APPENDIX B : SAMPLE OF LANGUAGE 'NUDES'

- STRUT -

- MAN -

- FIGURE MAN LFoot L Leg L Thigh Hips Chest Head L Arm R Arm R Leg R Thigh

- ELLIPSES LFoot 83 20 40
ELLIPSES L Leg 40 150 40
ELLIPSES L Thigh 80 150 60
ELLIPSES Hips 75 75 120
ELLIPSES Chest 100 230 130
ELLIPSES Head 80 120 70
ELLIPSES L Arm 40 135 40
ELLIPSES L Hand 40 135 40
ELLIPSES R Hand 40 175 40
ELLIPSES R Foot 40 20 40
ELLIPSES R Leg 40 150 40
ELLIPSES R Thigh 40 150 60
ELLIPSES R Arm 40 135 40
ELLIPSES R Hand 40 175 40
ELLIPSES Neck 30 60 30
ELLIPSES Nose 20 15 10

- JOINT R Knee R Leg 125 0 R Thigh 0 125 0
JOINT R Hip R Thigh 125 0 Hips 0 0 60
JOINT L Knee L Thigh 0 125 0 Hips 0 0 60
JOINT WAIST Hips 75 0 Chest 0 155 0
JOINT THroat Head 0 155 0 Neck 0 10 0
JOINT SPINE Neck 0 10 0 Shoulders 0 40 0
JOINT L Shoulder Shoulders 0 145 0 L Arm 0 125 0
JOINT R Shoulder Shoulders 0 0 145 0 RArm 0 125 0
JOINT LElbow L Arm 0 115 0 LLArm 0 115 0
JOINT RElbow RArm 0 115 0 RArm 0 115 0
JOINT WRIST PLArm 0 125 0 RHand 0 50 0
JOINT L WRIST LLArm 0 125 0 LHand 0 50 0
JOINT THORAX Shoulders 0 0 Chest 0 155 0
JOINT FACE Head 90 0 0 Nose 0 0 0

- GROUND 0.07
- VALBY 0 0 1660 46
- VIEW 0 0 1660 46
- PARCEL LFoot 50 3
BEND 0 0 RAnkle R Foot 0 15 3
BEND 0 0 RKnee R Leg 0 15 3
BEND 0 0 RHip R Thigh 0 20 3
BEND 0 0 WAIST Hips 0 23 3
BEND 0 0 L Hip Hips 0 90 3
BEND 0 0 LWrist L Arm 0 20 3
BEND 0 0 L Knee L Thigh 0 20 3
BEND 0 0 L Leg 20 3
BEND 0 0 THORAX CHEST 30 3
BENDBY LSMLDR SHLDRS -25 3
BENDBY LELBOW LUARM -10 3
DRAWAL 5
REPEAT 2
*
LOOP
RTALEL RFOOT -20 3
BENDBY RANKLE RFOOT 25 3
BENDBY RKNEE RLEG -40 3
BENDBY RHIP RTHIGH 35 3
BENDBY LMIP LIPS -30 3
BENDBY LKNEE LTHIGH 60 3
BENDBY LANKLE LLEG 10 3
BENDBY THORAX CHEST 30 3
BENDBY RSHLDR SHLDRS 5 3
BENDBY RELBOW RUARM -20 3
BENDBY LSHLDR SHLDRS -30 3
BENDBY LELBOW LUARM -15 3
GRNDAL
DRAWAL 6
REPEAT 2
*
*
CHANGE TO OTHER FOOT -
*
LOOP
RTALEL LFOOT -10 3
BENDBY LANKLE LFOOT 5 3
BENDBY LKNEE LLEG 5 3
BENDBY RKNEE RTHIGH 5 3
BENDBY RANKLE RLEG -10 3
BENDBY THORAX CHEST 10 3
BENDBY LELBOW LUARM -10 3
BENDBY RSHLDR SHLDRS 10 3
BENDBY RELBOW RUARM -15 3
GRNDAL
DRAWAL 2
REPEAT 2
*
*
CHANGE TO OTHER FOOT -
*
LOOP
RTALEL LFOOT -5 3
BENDBY LANKLE LFOOT -20 3
BENDBY LKNEE LLEG 30 3
BENDBY LMIP LTHIGH -5 3
BENDBY RHIP MIP -5 3
BENDBY RKNEE RTHIGH -65 3
BENDBY THORAX CHEST -30 3
BENDBY LSHLDR SHLDRS 15 3
BENDBY RSHLDR SHLDRS -15 3
BENDBY RELBOW RUARM 15 3
REFERENCES


(5) Phillipa Cullen, private communication 1972


(7) MOORE, A., "Modern Ballroom Dancing"

(8) VAGANOVA, A., "Basic Principles of Classical Ballet", Dover, NY 1969

(9) CAUSLEY, M., "An Introduction to Benesh Movement Notation", Max Parrish, London 1967


(11) TOSCAS, G., "Instant Animation from a Magic Box", International Photographer, Feb. 1971

(12) SHIGERU OHE, Film "Computer Animation", Moore School of Electrical Engineering, Uni of Pennsylvania, 1969


(14) BENESH, R. and McGUINNES, J., "Benesh Movement Notation and Medicine", Physiotherapy 60 (6), 1974, pp.176-178


(16) MENDO, G., private communication 1975

(17) KEEN, J., "Movement", Bassar Dept of Computer Science, Sydney University, 1973


