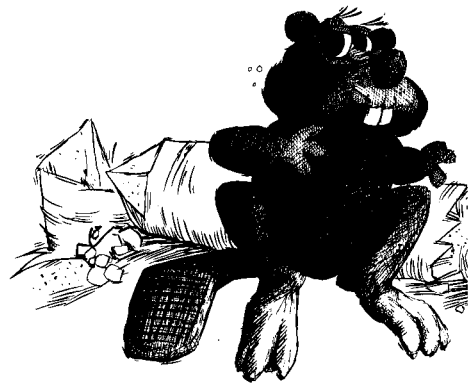
UNIVERSITY OF WATERLOO
UNIVERSITY OF WATERLOO
UNIVERSITY OF WATERLOO
COMPUTER SCIENCE DEPARTMENT
COMPUTER SCIENCE DEPARTMENT
COMPUTER SCIENCE DEPARTMENT



Animation of the Human Figure

Research Report

Don Herbison-Evans

CS-86-50

November, 1986

Computer Animation of the Human Figure Using Benesh Notation

Don Herbison-Evans

Department of Computer Science
university of Waterloo
and
Basser Department of Computer Science
Univerity of Sydney

ABSTRACT

This paper describes a system for producing animated dancing figures from a specification in Benesh Movement Notation. The system extends some of the techniques developed in a previous project by Politis at The University of Sydney, which interpreted 45 Benesh signs. The new system uses the ChoreoScribe editor at the University of Waterloo, and interprets 226 of the 2,200 signs handled by the editor.

Solutions to problems presented by the feet, elbows and knees, head and torso, body facing, and subbeats are described.

The display techniques available include stick figures, outline figures, outline figures with hidden lines removed, randomly perturbed monochrome shading, Floyd Steinberg monochrome shading, and full shaded colour, and can use a default figure or an arbitrary figure prepared using a NUDES script. These can be drawn on a variety of output devices, for plotting or frame buffer animation.

Appendices describe the NUDES script syntax, the requirements of an arbitrary figure for the interpreter, the programs used by the interpreter system, and how to use the commands for creating a display file ("nudes") and for showing it ("show").

data file containing not only the syntactic information of the signs and their positions on the stave, but also the semantic information: to what part of the body or semantic category each sign refers.

The human figure can be conveniently modelled on a computer using ellipsoids (6). Each body part then needs only nine numbers to describe it (the 3 axes lengths, 3 coordinates of the centre, and 3 orientation angles). Optionally, this information can be augmented either by 3 color components or the name of a picture file to be mapped onto its surface. A computer animation system, NUDES (Numeric Utility Displaying Ellipsoid Solids) modelling human figures this way, was developed at the University of Sydney. It can optionally display the figure in stick form, drawn outlines, grey scale or full shaded color.

This paper describes a set of programs for converting the output of ChoreoScribe into a suitable form for input to NUDES. It follows on an initial project at the University of Sydney in which an editor/interpreter pair were written accommodating the 45 basic signs of Benesh Movement Notation(12). The current interpreter has been expanded to accommodate 226 signs. ChoreoScribe handles 2,200 signs. Benesh Movement Notation has been estimated to use 52,000 signs (2).

The basic technique used in the interpreter is to take the frames of Benesh notation as key frames for animation. In order to do this, a variety of problems have to be solved.

The Figure

The body itself has to be dimensioned. For input to the NUDES display software, the body parts are ellipsoids. Initially, values are needed for the axis lengths of the ellipsoids making up the body, as well as the coordinates of the centre of each one, the topology of the joints between them, the coordinates of the joints. and possibly information on the coloring or texturing of each ellipsoid. These values are conveniently read from a file which is itself prepared by using a NUDES script, (see Appendix A). If no file is specified, the program uses default specifications of a female human figure in a leotard, complete with nose, hair in a bun, a feminine bosom, and hands with thumbs. This choice simplifies the understanding of the images, so that the orientation of the torso, head and hands can be readily discerned, e.g. see figure 4. An example of the use of a figure defined by a NUDES script is shown in figure 7.

A problem to be solved is that of matching of corresponding parts of the body between the interpreter and the file specifying the figure. Currently this is done by requiring that in the NUDES script, the first seventeen ellipsoids nominated in the first FIGURE declaration and the first sixteen joints be in a particular order (see Appendix B).

Other ellipsoids and joints can be specified in the NUDES script, including static ones not joined to the figure. These can be used for background and scenery. The feet must be defined with their long axes being y (as though the figure were on point).

Preprocessing

Each frame of a written Benesh score normally only contains the signs for those body parts that have moved since the previous frame. This fits the needs for animation conveniently, as the current state of the body model only needs updating for those parts that have moved.

The data files of ChoreoScribe maintain full information at each frame for 22 semantic entities, including hands, elbows, feet, knees, head, torso, hips, direction faced, subbeats and comments. ChoreoScribe maintains a list of signs and their stave positions in each of these semantic categories, copying those from the previous frame if a part has not moved. This redundant information is not needed by the interpreter, and a preprocessor was written to remove it.

This preprocessor also solves another problem: double versus single foot signs. In Benesh notation, if the feet are apart, individual signs for each foot are used. If the feet are adjacent, then combined signs are used. The data files of ChoreoScribe maintain information on the double foot signs as a semantic category separate from the left foot signs and the right foot signs. These double and single signs can thus become inconsistent. The preprocessor restores consistency between these related categories.

Interpreter

The notation is related to a parallel projection of the figure onto a stave consisting of five horizontal lines: floor, knees, waist, top of shoulders, and top of head. The hands, feet, elbows and knees are notated at their projected positions on the stave. Signs at these positions are used to indicate whether a body part is in front of or behind the coronal plane.

The following signs used in Benesh notation are accommodated by the interpreter described here:

(a) hand, elbow, foot and knee signs (24 signs)

(b) signs when an extremity contacts another part (10 signs)

(c) signs for the combined foot positions of classical ballet (including cou-de-pied) (55 signs)

(d) signs for combined twists, tilts and inclinations of the head and torso (125 signs)

(e) facing signs specifying the orientation of the whole body (8 signs)

(f) timing signs for positions reached on subbeats of the rhythmic meter (6 signs)

The basic technique used in this interpreter is to infer the depth from the position of a sign, the fixed lengths of body parts and nature of the sign. The 3D coordinates of the distal and proximal ends of each body part are then used to calculate quaternion angles (7) of the distal end relative to the proximal end. This is done for all the body parts notated in a particular Benesh frame. Then, movements to those orientations are interpolated over a series of animation frames.

A number of complications make the process more complex than outlined

above. Some of these are associated with special signs which speed the process of notation, particularly for positions and movements commonly used in classical ballet. Others arise from more general considerations.

Scaling

The relationship between the position of a sign on the stave and the projected position of the corresponding body part is not constant. The stave lines are equally spaced, but the heights in the body which they represent (floor, knees, waist, top of shoulders, top of head) are not. This is allowed for in the interpreter by using piecewise linear interpolation. Actually, Benesh Notation assumes that for hand and elbow signs the lengths of the upper arm, and fore arm, and for foot and knee signs, the lengths of the upper leg and lower leg are equal to the vertical spacing between the staves lines between which the signs occur. This detail has not been addressed in the current interpreter.

The Feet

A number of problems are posed by the feet. Signs for the feet are placed just below, on, or just above the floor stave line to indicate standing on flat foot, demipoint or full point respectively. Thus the exact position of the foot sign must be used to set flags to extend the ankle appropriately. The position of the foot can then be corrected to contact the ground exactly, if the figure is not in an aerial position.

The extension of the ankle, in the case of a foot being on point is to make the long axis of the foot vertical. In ballet, this is true even if a fondu or plié on point occurs. This contrasts with the pointing of a foot which occurs in ballet by default as the foot is lifted from the ground. In this case, the long axis of the foot is bent at the ankle so as to be colinear with the long axis of the lower leg.

These cases can be treated by precomputing the quaternion angles of the foot relative both to the pelvis and relative to the lower leg for the three positions (flat, demi-point, point) and choosing the appropriate set and reference part. The choice is made according to position of the foot sign relative to the floor stave line.

A complicating factor is that when the feet are close together, a combined sign is used in Benesh notation. An initial pass is used to break these up, and turn them in single foot signs located at the centres of the two halves of the combined sign. This must be calculated very carefully for the cou-de-pied and coupé signs, where the combined sign is tilted at an angle, and the locations of the ends of the sign relative to the floor line give the information on ankle flexion.

When the figure is supported by only one foot, the actual lateral (x) coordinate of the sign for that foot is ignored by the interpreter. This is because it is conventionally written on one side (its own side) of the centre line of that Benesh frame, whereas in practice it must be vertically under the centre of gravity of the body for stability. The interpreter ensures that it is positioned in this manner.

Elbows and Knees

If these are bent, they are notated as such with appropriate signs. These can be inconsistent with the placement of the corresponding extremity, given the fixed length of the limbs. If there is a discrepancy, the projected angular direction on the stave from hip to knee, knee to foot, shoulder to elbow and elbow to hand are used to derive the directions of the body part.

If a joint is straight, its position is not marked. Interpolation of its position is simple.

Another placement for which Benesh has special signs, occurs when an extremity is in contact with some other part of the body , and the corresponding bent elbow or knee lies in the coronal plane. In this case, the location of the joint is not notated. It must be inferred from the position of the extremity sign, the position of the pelvis or shoulder, and the fixed lengths of limb parts.

Masking

Another problem of both hands and feet is that if the projected position of a hand or foot is the same as that of the corresponding elbow or knee, only the elbow or knee is written. This happens when the distal limb part (lower leg or lower arm) points in the sagittal direction. To solve this problem, the interpreter inserts a default hand or foot position if an elbow or knee sign is present, which is overwritten if an actual hand or foot sign is present.

If the hand or foot sign is masked and this default is used, the sign of the depth (z) component is chosen to be the same as that of the knees for the legs, and positive (in the direction the figure is facing) for the arms.

Head and Torso Signs

In Benesh notation, these are based on a tilted cross. The angle of the vertical component, the lateral asymmetry of the horizonal component, and the vertical placement of the horizontal component are used to notate three angles specifying the orientation of the part relative to its proximally adjacent part. These components can be used to derive the angles using appropriate scale factors. The orientation so deduced is reduced to quaternion angles. In doing so, the correct sequence of axes must be used. Benesh Movement Notation uses the notional sequence y, z, x (18).

Body Facing

Because the movements of the body parts are stored in the interpreter as orientations relative to the proximally adjacent part, the reorientation of part of or the whole body presents no problems. Benesh has eight facing signs corresponding to the sides and corners of the stage. These are the conventional directions faced by the body in classical ballet. The interpreter applies an appropriate rotation to the whole body when a frame with a new facing sign is encountered.

The moving figure may be viewed from any direction. The default view is from behind, as is conventional for Benesh Movement Notation. Other views may be requested by specifying the Cecchetti direction from which the figure is

being viewed. Assuming the figure is on a stage these are

1 - downstage right
2 - downstage left
3 - upstage left
4 - upstage right
5 - stage front
6 - stage left (prompt)
7 - backstage
8 - stage right (opposite prompt).

Subbeats

The interpreter by default interpolates nine extra animation frames between each pair of Benesh frames (giving ten in all), although this can be overridden by one of the "nudes" command parameters. The resulting animation is shown by default at twenty frames per second, although this can be overriden by a "show" command parameter. Thus the default interpolates between Benesh frames at the rate of two per second.

In dance, many movements happen between the musical beats. These are notated as occurring, for example, on "an", halfway between beats, or "dai di": at one third and two thirds of the time between beats. There is a semantic difference between notes in music, which represents durations, and frames in Benesh notation, which are reached at particular moments in time.

The interpreter implements the reaching of positions on subbeats by shortening the ten (or whatever) frame interpolation down to, for example, five and three frames for "an" and "dai" respectively. Note that it is easy to write nonsense in the notation by, for example, following a position to be reached "an" by another to be reached on "dai". The interpreter detects such errors and comments upon them.

The "nudes" command program allows optionally the automatic insertion of animation frame numbers in the lower left corner of the frames.

Display

The resulting figures can be translated either into plot files or into image files.

The plot files can be either of stick figures, or ellipsoid outlines with all lines showing, or ellipsoid outlines with hidden lines removed. The NUDES suite contains programs for converting these plot files for drawing on either a pen plotter, or a display with Tektronix 4010 protocols, or an Ikonas 3000.

The stick figures are obtained from the intermediate ellipsoid specification file by ignoring the two minor axes of each ellipsoid, and just drawing its major axis, (see figure 2).

The methods for producing ellipsoid outlines (see figure 3) and removing hidden arcs (see figure 4) have been described before (6, 19).

Image files can be generated either in black and white, or in full colour. A

number of half toning algorithms were tried for black and white. Perturbing the shading by random numbers before thresholding to black or white gives a tolerable representation for single images, (see figure 5). Interesting artifacts appear when such images are animated. If the same set of random numbers are used in every animation frame the figure appears to be translucent, moving over a fixed stippled background. If the random numbers are uncorrelated between animation frames, the figure sparkles, as though covered in sequins.

The best half toning algorithm tried was that of Floyd and Steinberg (8), (see figure 6). Again there are some undesirable artifacts when such images are animated, because as the figure moves, the dithering pattern of the background only changes in one quadrant rather than istropically.

Coloured images are produced with 8 bits each specifying the shade at each pixel. These can be reduced to one bit per colour using Floyd-Steinberg halftoning, allowing animation of the coloured images (9). The dithering artifacts appear less severe with the coloured images, perhaps because the eye averages them over the three colours.

Implementation

The system uses 36 programs (see Appendix C) which are implemented currently on a DEC VAX8600 running under Ultrix (a Unix look-alike). The user need only interact with four programs :

ben - the ChoreScribe editor for creating, modifying and storing scores.
bprint - for printing out a notated score from a ChoreoScribe data file.
nudes - used for generating an image file from a ChoreoScribe data file.
show - for displaying an image file.

These programs and most of the ones that they call are written in *C*, except the hidden line program which is in Fortran. Fortran is used here for speed, as the available *C* compiler uses double precision arithmetic for "float" variables, which is more than a factor of two slower than the single precision arithmetic used for "real" variables by the available Fortran compiler.

The "ben" commands invokes ChoreoScribe. Its use and function to create or modify a score have been described elsewhere (4).

The "bprint" command allows a ChoreoScribe data file to be printed out as a score on a variety of devices. It too has been described before (2).

The "nudes" command can be invoked with a series of flags, (see Appendix D). These can be in any order. The argument is the name of the ChoreoScribe data file. The flags allow a choice of viewing direction, figure representation etc. If no arguments are given, then "nudes" lists the first sixty lines of its own source code (via the program "more", giving one screen full at a time), which is Appendix D, describing the use of each of the flags. The "nudes" command can also be used to generate image files or figure files from NUDES scripts.

The time taken for the generation of an image varies according to the operations chosen (see table 1).

The time is dominated by the scan conversion or rendering section of the software, and so is approximately proportional to the number of ellipsoids in the figure.

The "show" program selects an appropriate free Ikonas frame buffer for frame buffer animation of the image, loads the image file into the buffer, loads an animation program into the controller, and runs it. The animation can be single stepped through the frames, or run at arbitrary speeds. The default is 20 animation frames per second. Its usage is also controlled by flags, which are in the initial comment block of the source code, and which are listed if the program is run with no arguments at all.

Animation

One of the Ikonas devices which is available in the Computer Graphics Laboratory at the University of Waterloo has 32 bit planes of 1024 x 1024 resolution. These can be variously mapped to the screen at resolutions of 512 x 512, 256 x 256 or 128 x 128, allowing frame buffer animation of 128, 512 or 2048 animation frames respectively, displaying sequences for 6, 25 or 102 seconds respectively, at the default frame rate. The "nudes" command program automatically selects one of these alternatives, and pads out the animation by repeating the first and last frames so that the frame buffer is fully utilized. Padding the first and last frame is pleasing to the eye, giving a proper beginning and ending to a movement sequence.

Acknowledgements

This work was supported financially by grants from the University of Sydney (Australia) and from the National Science and Engineering Council (Canada) through the Computer Graphics Laboratory at the University of Waterloo. Frame buffer manipulation software was kindly provided by Alan Paeth. Much programming assistance was provided by David Forsey and the other members of the Computer Graphics Laboratory at the University of Waterloo. Important contributions to the NUDES software were made by Danuta Kucharska and Peter Gummer at the University of Sydney and John Chapman at the University of Waterloo.

The inspiration for the project was provided by the late Phillipa Cullen, choreographer from Sydney, and continued by Rhonda Ryman of the Dance Group at the University of Waterloo. Without Rhonda's kind and patient teaching it would not have happened.

The author is also indebted for the hospitality shown at the University of Waterloo by Kellogg Booth, John Beatty and the staffs of the Computer Science Department and the Dance Group.

Thanks are also due to John Bennett, head of the Basser Department of Computer Science at the University of Sydney for his encouragement and many helpful suggestions for the computers and dance project over a period of several years.

References

(1) MacGuiness-Scott, J., "Movement Study and Benesh Movement Notation", Oxford University Press (1983).

(2) Ryman, R.S., "An Overview of the Benesh Editor Project", York Symposium, York University, Toronto (1986).

(3) Politis, G., and Herbison Evans, D., "Computer Choreology Project at the University of Sydney", Proc. Conf. Australasian Computer Graphics Association, pp. 86-89 (1983).

(4) Ryman, R.S., Dransch, D.O.K., and Beatty, J.C., "ChoreScribe: an "Intelligent" Editor for Chorelogists", The Choreologist, No. 31, pp. 13-24 (1985).

(5) Hughes-Ryman, R., and Ryman, R.S., "MacBenesh: A 'Word-Processor' for Benesh Movement Notation", The Chorelogist, No. 31, pp. 25-29 (1985).

(6) Herbison-Evans, D., "NUDES 2: A Numeric Utility Displaying Ellipsoid Solids, Version 2", Computer Graphics, Vol. 12, No. 3, pp 354-356 (1978)

(7) Herbison-Evans, D., and Richardson, D.S., "Control of Round-Off Propagation in Articulating the Human Figure", Computer Graphics and Image Processing, Vol. 17, pp. 386-393 (1981).

(8) Floyd R.W., and Steinberg, L., "An Adaptive Algorithm for Spatial Gray Scale", Society Information Displays, Int. Symp. Digest of Tech. Papers, pp. 36 (1975).

(9) Paeth, A.W., and Booth, K.S., "Design and Experience with a Generalised Raster Toolkit", Proc. Graphics Interface '86, pp. 91-97 (1986).

(10) Hutchinson Guest, A., "Dance Notation", Dance Books (London), (1984).

(11) Savage, G.L., and Officer, J.M., "CHOREO: An Interactive Computer Model For Dance", 5th Man-Computer Communications Conference, Calgary, Academic Press (London), pp. 223-249 (1977).

(12) Politis, G., and Herbison-Evans, D., "A Computer Graphics Interpreter for Benesh Movement Notation", Proc. 3rd Australasian Conference on Computer Graphics, Australasian Computer Graphics Association (Sydney), pp. 25-30 (1985).

(13) Brown, M.D., and Smoliar, S.W., "A Graphics Editor for Labanotation", Computer Graphic, Vol. 10, No. 2, pp. 60-65) (1976).

(14) Barenholtz, J., Wolofsky, Z., Ganapathy, I., Calvert, T.W., and O'Hara, F., "Computer Interpretation of Dance Notation", Computing in the Humanities, University of Waterloo Press (Waterloo), pp. 235-240 (1977).

(15) Smoliar, S.W., and Weber, L., "Using the Computer for a Semantic Representation of Labanotation", Computer Graphics, Vol. 17, No. 3, pp.

51-62 (1983).

(16) Mendo, G., and Archer, L.B., "A Study of Computer Choreology", Benesh Institute of Choreology (London) 1975).

(17) Piankoff, A., "The Pyramid of Unas", Princeton University Press, p. 89 (pl.64) (1968)

(18) Herbison-Evans, D., "Rotation Signs in Benesh Movement Notation" Technical Report CS 86-64, Department of Computer Science, University of Waterloo (1986).

(19) Herbison-Evans, D., "Hidden Arcs of Interpenetrating Ellipsoids", Australian Computer Journal, Vo.15, No.2, pp 65-68 (1983).

(29) Dransch, D.O.K., Beatty, J.C., and Ryman, R.S., "ChoreoScribe: A Graphics Editor to Describe Body Position and Movement Using Benesh Movement Notation", University of Waterloo, Department of Computer Science, Technical Report CS-86-48, p.111 (1986).

Table 1

stick figure skeletons	3
ellipsoid outlines with hidden lines visible	10
ellipsoid outlines with hidden lines removed	8
full shaded colour	20
black and white half toned using Floyd-Steinberg algorithm	30

Elapsed times in seconds per animation frame for generation of an animation file from a Benesh score, using a figure composed of twenty-three ellipsoids.

Appendix A

The Animation Command Language: NUDES
(Numeric Utility Displaying Ellipsoid Solids)

Introduction

The Numeric Utility Displaying Ellipsoid Solids (NUDES) language allows the production of figures and scenes composed of jointed figures made out of ellipsoids for input into the Benesh interpreter, and also for the production of sequences of animation directly of these moving figures. The figures, scene or animation required has to be specified in the form of a program in the NUDES language. This program has to be in the form of a file. The file must have a name ending in .n .

The contents of such a file may be thought of as having three parts:

(1) a set of comments for humans to read about the file

(2) list of items to be manipulated (declarations)

(3) list of manipulations (movements)

It is good practice to make the first line of the file a comment, (i.e. start with an asterisk on the left hand side) containing the name of the file. The next few lines should also be comments: indicating the author, date, and a brief description of the animation, and the range of frame numbers.

The last line of the file must be the line:

STOP

The list of items to be manipulated (the declarations) should contain commands specifying figures, ellipsoids, joints and variables.

The list of manipulations (the movements) should contain commands, each of which specify when it is to happen, how it is to be distributed over that period, what is to happen, to what items it applies, together with other necessary information such as direction.

A command line in any part of the file will typically have one of more KEYWORDS specifying the command, some names of various items, and some values. These words and values should all be separated from each other by one or more spaces on the line.

Sets of movements grouped into subroutines should be placed in the file after the main program, and before the STOP command. The subroutines can be placed in any order.

Declarations

The fundamental type of item is an ellipsoid. Ellipsoids may be jointed together to give figures. The declaration of a figure should occur before the declarations of the ellipsoids in the figure. The ellipsoids should be declared before the declarations of the joints between those ellipsoids.

Figures are declared by a line starting with the word FIGURE, followed by an arbitrary name (of the figure), followed by the number of ellipsoids in that figure, followed by that many names of ellipsoids. If the figure has left/right symmetry, the ellipsoids should be in pairs with the left one before the right one, and such pairs should be listed before any unpaired ellipsoids.

An ellipsoid is declared by a line starting with the word ELLIPSOID, followed by the name of the ellipsoid, followed by half the width of the ellipsoid (left to right), half the height, and half the depth. Ellipsoids may only be declared initially with their major axes aligned with the screen axes. Other orientations may be obtained by specifying actions over the time interval between frames zero and one (frame zero is never shown). The lengths of the semi-axes of ellipsoids should be chosen in relation to the size of the screen, which is 1000 units wide. It is often convenient to make figures an arbitrary size, and use a GROFIG action between frames 0 and 1 to scale them for screen units.

A joint between two ellipsoids is declared by a line beginning with the word JOINT followed by the name of the joint, followed by the name of one of the ellipsoids being joined, then the distance of the joint from the centre of that ellipsoid (as three components: to the right (x), up (y), and away (z)), then the name of the second ellipsoid, and the distance of the joint from its centre (as three components). Only two ellipsoids may meet at a joint. Ellipsoids may not be joined so that they form any sort of closed ring.

A number of figures have already been generated over the years for use in NUDES programs. Much work can be saved by using the declarations of such a figure from an existing file. The Editor can be used to change the actions and the initial comment lines of a copy of such a file.

Movements

A movement specification starts with how the movement is to be distributed over the frames in which it is active. It may be done for every frame (REPEAT), or spread uniformly over the frames (LINEAR). It may accelerate from rest (ACCELERATE) or decelerate to end at rest (DECELERATE). It may go through an accelerate/decelerate cycle (QUADRATIC). This last most naturally emulates muscle and inertial movement.

After the distribution come the frame numbers of the start and finish of the action. These may be the names of variables containing the numbers.

Then comes the name of the required action.

Actions

Many actions require the specification of a direction. This is done by specifying an axis (x,y or z) of some reference ellipsoid. This can be the ellipsoid that the action affects, or an adjacent one, or any other ellipsoid. One special ellipsoid WORLD, is provided which is aligned with the screen axes. (x=to the right, y=up, z=away).

Many actions can be applied to just a single figure which is nominated in

the command. Alternatively, ALL can be put at this point to refer to all the figures in the scene.

Some actions specify how you want things to end up. Others specify what incremental change is desired of the current position. The incremental commands are the easier to use directly. The others are useful if desirable positions or orientations have been stored in variables with a previous CENTRE, ANGLES or LINK command.

WRITE

Write out a file of a scene for input to the Benesh interpreter.

COLOUR (ename) (redvalue) (greenvalue) (bluevalue)

Assign red/green/blue components of colour to an ellipsoid.

TEXTURE (ename) (iname) (xoffset) (yoffset)

Assign image file name to be texture mapped onto an ellipsoid with given offsets for the mapping from the origin in the image to the major x semiaxis extremum of the ellipsoid.

GROFIG (fname) (ename) (xfactor) (yfactor) (zfactor)

Scale given figure about the centre of given ellipsoid, multiplying all parts of the figure by given factor.

GROELL (ename) (xfactor) (yfactor) (zfactor)

Scale a single ellipsoid in size keeping all joints fixed.

GROJNT (ename) (jname) (xfactor) (yfactor) (zfactor)

Scale a single ellipsoid in size keeping a nominated joint of it fixed, and allowing its other joints and connected ellipsoids to move appropriately.

GROUND (fname)

Make lowest point of given figure touch ground plane.

MOVETO (fname) (ename) (x) (y) (z)

Move given figure so that centre of given ellipsoid is at given position.

MOVEBY (fname) (referenceellipsoid) (x) (y) (z)

Move given figure by given amounts parallel to the axes of given reference ellipsoid.

SPINTO (fname) (ename) (referenceellipsoid) (colatitude) (longditude) (twist)

Spin given figure about centre of given ellipsoid, rotating given ellipsoid to given eulerian angles relative to axes in the reference ellipsoid.

SPINBY (fname) (ename) (referenceellipsoid) (angle) (axis)

Spin given figure about centre of given ellipsoid, rotating by given angle about given axis in the reference ellipsoid.

BENDTO (movingellipsoid) (jname) (referenceellipsoid) (colatitude) (longitude) (twist)

Bend figure at given joint, rotating given ellipsoid to given eulerian angles relative to axes in the reference ellipsoid.

BENDBY (movingellipsoid) (jname) (referenceellipsoid) (angle) (axis)

Bend figure at given joint, rotating given ellipsoid and all ellipsoids joined to it by given angle about given axis in the reference ellipsoid.

FLEX (movingellipsoid) (jname) (angle)

Bend body part about x axis of linked ellipsoid.

ROTATE (movingellipsoid) (jname) (angle)

Twist body part inwards about y axis of given ellipsoid.

ABDUCT (movingellipsoid) (jname) (angle)

Bend body part away from body about z axis of linked ellipsoid. Note that rotate and abduct both rely upon the order in which ellipsoids were declared during each figure definition to decide which direction to bend or twist a body part. Odd- numbered ellipsoids are taken to be on the left-hand side of the body; even-numbered ellipsoids are taken to be on the right.

DETACH (movingellipsoid) (jname) (fname)

Break a figure into 2 figures at the given joint, naming the figure containing the given ellipsoid by the given name, and keeping the old figure name for the other part of the old figure.

ATTACH (ename) (jname) (ename) (x) (y) (z)

Join two figures together making a joint at the distance specified from the centre of the second ellipsoid. Note that the use of detach and attach may disrupt the ability of rotate and abduct to tell left from right.

SET (variablename) (anything)

Set a named variable to a value.

NEGATE (variablename)

Change the sign of a variable.

ADD (variablename) (x) (x)

Add two values together.

SUBTRACT (variablename) (x) (x)

Subtract second value from first.

MULTIPLY (variablename) (x) (x)

Multiply two values.

DIVIDE (variablename) (x) (x)

Divide second value into first.

ANGLES (ename) (referenceellipsoid) (variablename) (variablename) (variablename)

Store the orientation angles of an ellipsoid.

CENTRE (ename) (variablename) (variablename) (variablename)

Store the coordinates of the centre of an ellipsoid.

AXES (ename) (variablename) (variablename) (variablename)

Store the lengths of the semiaxes of an ellipsoid.

LINK (jname) (variablename) (variablename) (variablename)

Store the coordinates of a joint.

OBSERVE (angle) (angle) (angle)

View the scene from the specified direction.

PLACE (x) (y) (z)

Make the specified position the centre of the scene.

Special Commands

The VIEW command is followed by a pair of frame numbers. Only frames in that range will be generated, starting with the frame after the first one nominated.

The SPEED command allows the interpolation of extra frames if its parameter is positive, and the deletion of frames if negative. Thus

SPEED 5

causes four extra frames to be interpolated between every pair of frames, making it five times longer. This can be an easy way of lengthening a film and smoothing the actions.

The command:

SPEED -5

causes only every fifth frame to be displayed, starting with frame 1 (i.e. frames displayed will be 1, 6, 11, 16, 21 etc.).

The SPEED and VIEW commands can both be used at the same time. The frame numbers in the view command are not affected by a positive speed parameter. A negative speed parameter can give confusing effects, e.g.

SPEED -5

VIEW 12 15

shows no frames, as none of the frames allowed by the speed command are in the view time period.

The speed and view commands can be anywhere in the file. If there is more than one of either, the last one is the one used. I recommend putting only one of each, and putting these immediately after the initial title comment line.

Applying Actions to Various Figures

An action or set of actions can be written in terms of variable names for the ellipsoids, joints, etc. This can be used for a variety of figures by preceding its use by a REPEAT of a CALL to a subroutine that assigns the appropriate names for the figure to those variables.

This technique can also be used to apply the same set of actions first to the left part of a figure then to the right. Before the action is placed, a CALL to one of two subroutines can be made, one placing the left body parts in the variables, the other placing the right body parts.

Appendix B

The first seventeen ellipsoids in the first FIGURE declaration in a NUDES script specifying a figure or scene for the Benesh interpreter should be equivalent respectively to:

left foot
right foot
left lower leg
right lower leg
left thigh
right thigh
left upper arm
right upper arm
left lower arm
right lower arm
left hand
right hand
pelvis
chest
shoulders
neck
head

and the first sixteen joints should be in order equivalent respectively to:

left ankle
right ankle
left knee
right knee
left hip
right hip
left shoulder
right shoulder
left elbow
right elbow
left wrist
right wrist
waist
thorax
throat
cervical spine

Appendix C

programs used from /p/nudes/bin/ -

binter	interpret a benesh data file into a nudes.3 file
clean	remove incorrect data from a benesh data file
complw.b	lexically analyse nudes.n script producing nudes command file
fradd	add frame numbers to readable plot file
hideu.b	project ellipsoids to ellipse outlines, removing hidden lines
im0032	load and compact 32 animated frames using an Ikonas
im0128	load and compact 128 animated frames using an Ikonas
im0512	load and compact 512 animated frames using an Ikonas
im2048	load and compact 2048 animated frames using an Ikonas
imt0032	load and compact 32 Tektronix files using an Ikonas
pada	add extra first and last frames to an ellipsoid file
plotel	produce readable plot file from ellipse outlines
plplot	convert readable plot files into binary unix plot files
prfrmz	produce ellipsoid file from nudes command file
prmess	write out a diagnostice message from a prfrmz error file
skela	draw major axis skeletons of ellipsoids
tekpl	produce Tektronix file from readable plot file
raselp	produce full colour image(s) of ellipsoids
rasq	produce shaded dithered 1 bit image(s) of ellipsoids
rasta	produce Ikonas image(s) from readable plot file
vim0032	load and compact 32 animated frames using memory
vim0128	load and compact 128 animated frames using memory
vim0512	load and compact 512 animated frames using memory
vim2048	load and compact 2048 animated frames using memory
visib.b	project ellipsoids to give ellipse outlines

programs from other directories-

/p/im/bin/imikr	read an image from an Ikonas
/p/im/bin/imikw	load an image into an Ikonas
/usr/graphics/ikonas/ikfree	release an Ikonas
/usr/graphics/ikonas/iknuke	clear an Ikonas
/usr/public/iktek	load a Tektronix file into an Ikonas

programs for preparing data-

/p/benesh/ben	ChoreScribe editor for Benesh notation

/p/benesh/bprint	print a ChoreScribe data file as a score

programs for using results-

/p/nudes/bin/show	display animated images on an Ikonas
/p/im/bin/imikw	load an image into an Ikonas
/p/im/bin/imikcycle	cycle through animation frames on an Ikonas
/usr/public/hptek	make a HP2448a display work like a Tektronix
/usr/bin/hpplot	draw a unix plot file on a HP 7221C plotter

Appendix D

```
/*      nudes

   runs the suite of nudes programs.

  usage-

    nudes file [output] [representation] [options]

  allows choice of input file :-
    searches for  (a) file (of Benesh data) in this directory
    if none, then (b) file (of Benesh data)
                      in local subdirectory benesh_data
    if none, then (c) file (of Benesh data) in a /p/benesh/benesh_data
    if none, then (d) file.n, file.4, file.3 or file.2 (nudes files)
                      in this directory.

  allows choice of output :-
    -t   Tektronix plot file
    -p   Unix plot file
         (for Hewlett-Packard plotter, or Versatec or Imagen printers)
    -a   alphanumeric output
    -4   nudes numeric command file
    -3   nudes ellipsoid specification file
    -2   nudes ellipse specification file
    -i   Ikonas image file (default)
         (choosing appropriate number of frames
          viz. 1, 32, 128, 512, 2048 )

  allows choice of figure representation :-
    -s    stick figure skeletons
    -v    ellipsoid outlines with hidden lines visible
    -h    ellipsoid outlines with hidden lines removed (default)
    -r    half-toned with correlated randomness in all frames
    -u    half-toned with uncorrelated randomness
    -f    half-toned by Floyd-Steinberg algorithm
    -c    full shaded colour

  allows various options :-
    -x    suppress the viewing x rotation in binter
    -e n  extra (n-1) frames to be inserted between
          each pair of Benesh frames
          (default n=10)
    -k    keep intermediate files and temporary directories
          (default: delete intermediate files and directories)
    -n    do not insert frame numbers
```

(default: insert frame numbers for s,v,and h representations)
-d n debugging mode: 'n'=1: system calls; 'n'<1: everything
(default: 2, i.e. none output)
-b n begin showing the figure from frame 'n' of a benesh score
(default: 0, i.e. the beginning)
-l n show sequence of length 'n' frames of a benesh score
(default: the full length or 204, whichever is smaller)
-g f get a figure from file f
-m use main memory rather than an Ikonas to assemble animation
-w n view figure, by looking from given Cecchetti wall.
The figure is facing downstage, i.e. Cecchetti wall 5
For other viewing directions, 'n' should be:-
1 - look from the downstage stage-right corner
2 - look from the downstage stage-left corner
3 - look from the upstage stage-left corner
4 - look from the upstage stage-right corner
5 - look from downstage
6 - look from stage left
7 - look from upstage (default)
8 - look from stage right

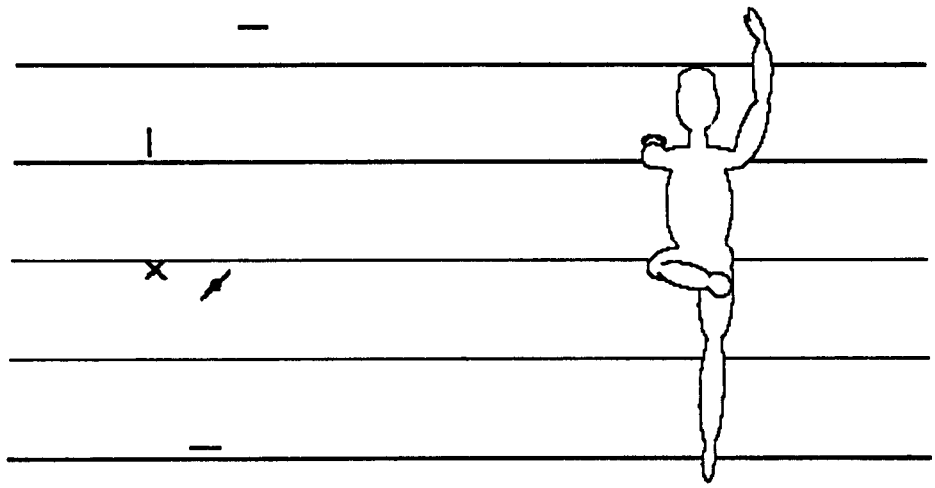


Figure 1 Relationship of figure to Benesh stave and signs.

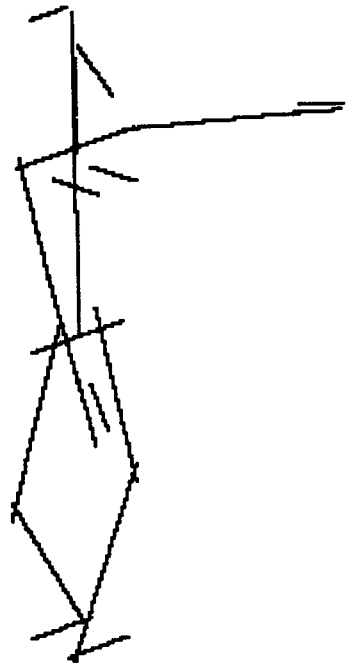


Figure 2 Stick figure representation, using major axes of the ellipdoids.

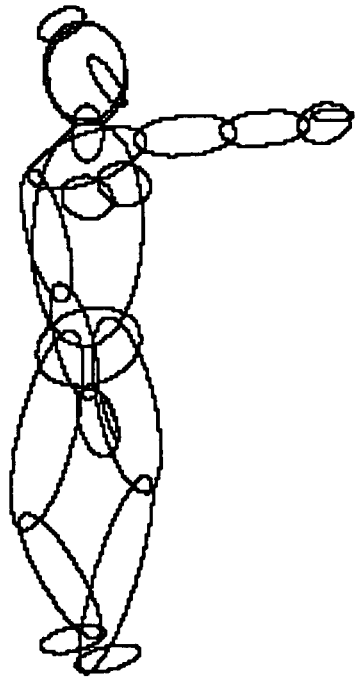


Figure 3 Ellipsoid outline figure.

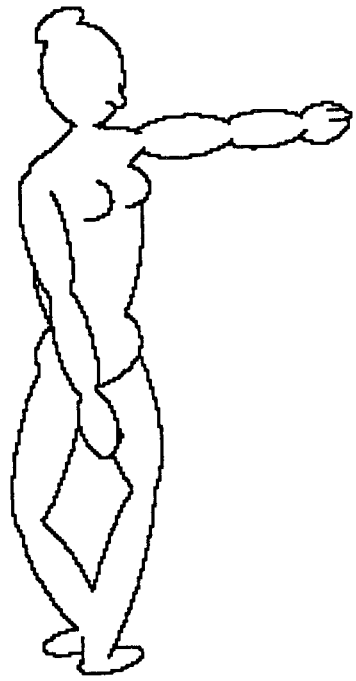


Figure 4 Ellipsoid outline figure with hidden lines removed

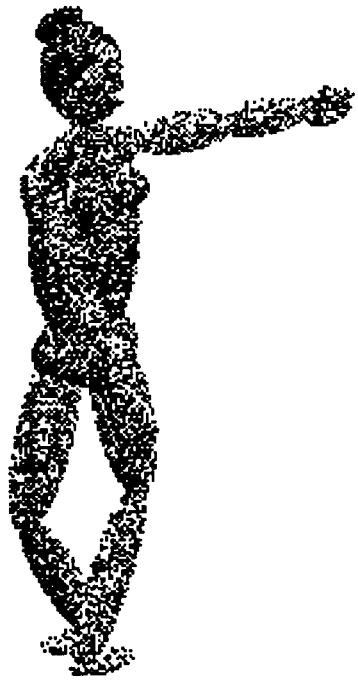


Figure 5 Shaded figure using random perturbation of shade value before thresholding.

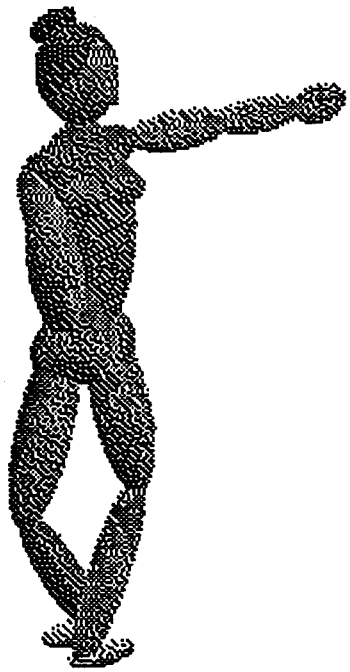


Figure 6 Shaded figure using Floyd-Steinberg error propagation algorithm.

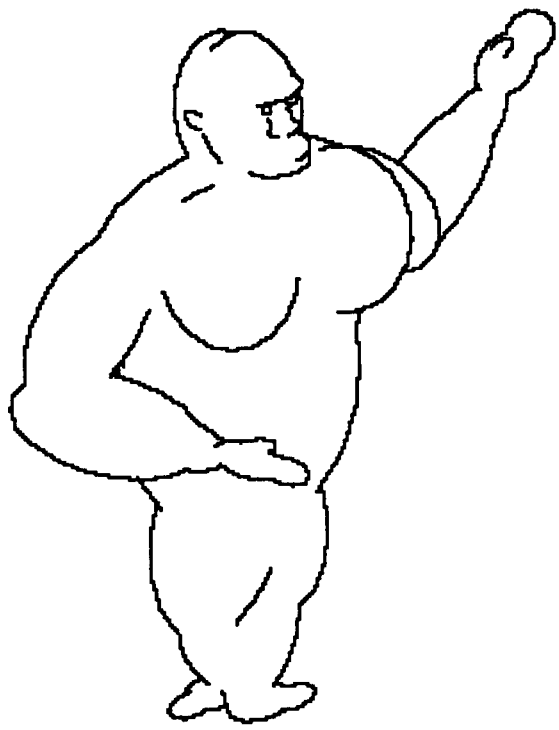


Figure 7 Non-human figure dancing, prepared from a NUDES script.

Computer Animation of the Human Figure Using Benesh Notation

D. Herbison-Evans

Introduction

Describing the movements in time and three dimensional space of an articulated figure has always been a problem. The longest history of attempts to solve it is in the field of dance (e.g. 17).

Over the centuries, a number of dance notations have been proposed (10), to solve this problem. They typically use a large vocabulary of graphic signs distributed over a two dimensional staff. In order to assist with the calligraphy of the signs, and the problems of updating a notation score, several of the notations these have been the subject of computerisation. Computer Editors have been created for subsets of Laban (13, 14), Massine (11) and Benesh (2, 3, 5) notations. In order to assist with the understanding and verification of a score, a computer interpreter might be used to draw an animated figure performing the scored movements. Computer interpreters have been proposed for Laban (15) and Benesh (16) notations, and actually written for subsets of Laban (14), Massine (11) and Benesh (12) notations.

The work described here is a pilot study designed to assess the problems of bringing two of these projects together (4, 12). Some of the serious problems of interpreting Benesh notation on a computer were solved in the precursor project (12). The resulting interpreter has now been re-written to utilise the data files of much more extensive editor (4). This work describes some of the problems encountered in this process, and their solution. The eventual aim is to create an editor/interpreter pair to cover the use of a broad enough subset of Benesh notation to allow the notation and interpretation of arbitrary movements of a single figure.

Benesh Movement Notation

Benesh Movement Notation is a graphical language for describing general human movement (1). It is used professionally in physiotherapy and in dance. For example, it is used by the major world ballet companies for the maintenance of their repertoires (e.g., Royal Covent Garden Ballet, Canadian National Ballet, Australian Ballet).

Part of the the language consists of a set of signs drawn on a 5 line stave similar to that used for music. Some of these signs are in groups called 'frames'. Each frame shows a symbolic snapshot of the figure at a particular moment in time. For an example, see figure 1.

The University of Waterloo, an editor, "ChoreoScribe," has been written for a subset of Benesh notation. ChoreoScribe is an intelligent editor, having a