From Dance Notation to Human Animation: The LabanDancer Project

Lars Wilke
Credo Interactive Inc
lars@charactermotion.com
www.charactermotion.com

Tom Calvert
Simon Fraser University
tom@sfu.ca
www.siat.sfu.ca

Rhonda Ryman
University of Waterloo
rsryman@healthy.uwaterloo.ca

Ilene Fox
Dance Notation Bureau
IleneFox@dancenotation.org
www.dancenotation.org

Abstract

Symbolic systems such as Labanotation for notating dance and choreography provide a critical tool for the preservation of cultural heritage in what once was considered an "illiterate" art form. While the goals of such notation systems are laudable, the unfortunate reality is that most dancers and choreographers cannot read or write the notation; that is, they are loath to take the considerable effort to learn a rich, but complex methodology. To make Labanotation scores more accessible the LabanDancer system has been developed to translate Labanotation scores recorded in the LabanWriter editor into 3-d human figure animations. A major challenge in the development of this translator has been to find approaches that are general enough to create reasonable animations for a wide variety of different movements. Any translator must also take account of the context of a movement since this can affect the interpretation of the Labanotation scores.

Keywords: Computer graphics, human figure animation, dance, notation.

1. Introduction

"The dance is unhappily an illiterate art; it knows nothing of yesterday; it cares nothing for tomorrow."

(New York Times, Dec. 4, 1932)

Before the advent of movement notation systems, dance was considered an ephemeral art. Notation allows objective documentation of dance in the same way that a musical score allows a composer to specify the intent of a musical composition. While a number of dance notation systems exist, the most common are Labanotation and Benesh notation. The former is the most common in North America and the Dance Notation Bureau (DNB) in New York has been working for 63 years to create, house and disseminate dance scores recorded with this system (www.dancenotation.org). The resulting archive provides scholars, students, performers and the public with an easily accessible, detailed record of choreography that allows the study of the dances in a way that is not possible with any other medium.

Although notation is fundamental to dance reconstruction and to dance research, unfortunately very few dancers or choreographers can read dance notation and even fewer are capable of producing scores. Thus, many have recognized that there would be a considerable advantage to have a computer based tool to animate the notation. This impetus can be traced back as far as the 1960's with Noll's article in Dance Magazine [1] and Merce Cunningham's discussions about the same time [2]. Probably the first attempt to apply computers to Labanotation was Zella Wolofsky's 1974 Simon Fraser University masters thesis on the interpretation of selected Labanotation commands [3]. Subsequently there were a number of projects that focused on different aspects of the interpretation of Labanotation [4,5,6].

Concurrent with the development of computer based approaches to interpretation of

Labanotation, there were parallel efforts to realize systems to assist in the creation and editing of dance notation scores and other tools to specifically assist choreography. LabanWriter was developed at the Ohio State University under the leadership of Lucy Venable [7] and at Simon Fraser University Tom Calvert lead a team developing a variety of human animation techniques [8]. LabanWriter took advantage of the graphics capabilities of the then relatively inexpensive Macintosh computer to provide a simple and intuitive word processor like system for creating and editing Labanotation scores. MacBenesh - a similar system for Benesh Notation - was developed by Rhonda Ryman and her colleagues at University of Waterloo [9]. At about the same time Life Forms (www.danceforms.com) was developed to provide choreographers and animators with a simple, user friendly system to experiment with patterns of movement in animated human figures. Students, notators, educators, and choreographers have been using both LabanWriter and Life Forms and many have suggested that they should be linked. Details of other efforts to create notation, to edit it and to animate it can be found elsewhere [10,11,12,13].

This paper describes the development of LabanDancer, a tool for animating Labanotation. The input to LabanDancer is a digital score from the LabanWriter application for composition and editing of Labanotation. In developing this new interpretation system we have sought out general techniques that handle the great majority of movement situations correctly and that recognize the context for the movement.

2. Labanotation

In 1928, Rudolf Laban, an eminent choreographer and dancer in his era, first published Kinetography Laban, a notation system for representing human motion which later became known as Labanotation. The system is a precursor of modern keyframe animation systems in the sense that it captures positional information for various body parts (channels) at distinct points in time.

Poses or Gestures for a particular limb are defined by one of nine direction shapes written on the page plus the Level, indicated by the fill pattern of the shape. The length of the shape symbol defines the length of time it takes to perform the gesture. The symbols are arranged on a Staff which takes its inspiration from its musical namesake. It is read from bottom to top as opposed to left to right; the double bars at the bottom indicate the start of the motion sequence; those at the top indicate the end. Symbols found below the bottom double bars define the starting pose. As in a musical score, the staff is divided into measures, the time lengths of which are defined by the tempo and the number of beats per measure. One or more staves can be arranged into a Score, each staff representing the motion of an individual dancer. Figure 1 shows an example score for a single dancer.

In the horizontal direction, the staff is divided into columns that group symbols into the body parts which they affect. The columns are arranged about the centre line of the staff to represent the left-right symmetry of the body.

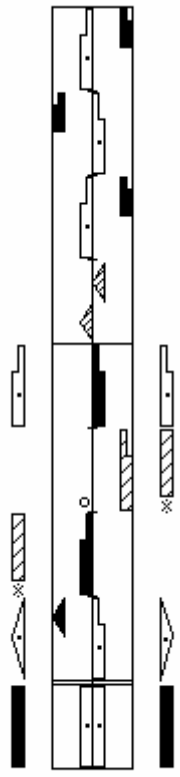

Figure 1: A Labanotation staff

Labanotation uses the symbols in the two most central columns in a distinct way. Here, the symbols describe the direction of the transfer of weight from one supporting limb (usually a foot) to another, thus defining the locomotion of the subject. The level indicates whether the step is done with bent knees, straight legs, or on point. As with gesture symbols, the length of the symbols indicates the amount of time to perform the step. Gaps between symbols in the vertical direction mean the subject has left the ground, as in running, hopping or leaping.
Labanotation provides a vast array of ancillary symbols that can be used to modify the meaning of the main symbols, measures or columns. Additional symbols are used to indicate the arrangement of the subjects on the stage (the Floor Plan) and the paths they take, or to indicate spatial relationships between entities (limbs or subjects). Special symbols are used to indicate that a direction or turn symbol affects a limb other than the one implied by its column position. Labanotation can thus provide an arbitrary amount of prescriptive detail. It can be used equally well to quickly sketch out a movement in broad brushstrokes or to define a motion right down to the position of the sub-joints on a finger. A complete description of the Labanotation system can be found in Ann Hutchinson-Guest's book [14] and tutorials are available online (http://www.dancenotation.org).

2.1 The LabanWriter Application

LabanWriter is a Macintosh based 2D graphics editor, specialized for creating Labanotation scores. Professor Lucy Venable of the Ohio State University Dance Department led the development of LabanWriter with team members Scott Sutherland, David Ralley and George Karl. (It is freely available for download from http://www.dance.ohio-state.edu/labanwriter). The user creates one or more staves on the LabanWriter virtual page, and selects from a palette of Labanotation symbols. The symbols are placed on a staff in the column representing the appropriate body part. The duration of direction and turn symbols is indicated by their vertical length which can be stretched or squeezed by the user. Modifier symbols are placed in proximity to the main symbols or columns that they modify – this is an inherently error prone operation.

The Labanotation symbols used in LabanWriter are treated as 2D graphical objects that fall loosely into two subclasses – stretchable and fixed size. Stretchable symbols, such as those for direction and turns, have associated time durations, while fixed size symbols normally modify or annotate columns, measures or other aspects of the score. As stored in LabanWriter, the symbols do not have explicit column and timing information - the Cartesian position of symbols is used to deduce their column and measure. Thus, the ASCII LabanWriter file is strictly for graphics layout on the screen, and contains no knowledge about the structure of Labanotation, dance or human movement. A detailed description of the LabanWriter file format can be found at: http://www.dance.ohio-state.edu/labanwriter/LW4/LW44FileFormat.html. It will be noted that each record in the file contains all information necessary to draw the symbols - type, position, size, colour, special fonts etc. Within the data stream the information is in the order that it was laid down by the user – it is not sorted in any way. The Labanotation score can be printed by LabanWriter, or stored as a 2D raster image in one of three popular graphic file formats (PNG, PICT or JPEG). This graphics file is used by the Windows version of LabanDancer as a means of displaying the score.

3. LabanDancer

LabanDancer is the name of the stand-alone application we have developed to translate LabanWriter files into animation for a single human figure. It contains an implementation for all of the algorithms required for the translation. These are described below.

3.1 Parsing LabanWriter Symbols into a Composite Score

To translate LabanWriter files into animation it is first necessary to convert the stream of graphical symbols into a set of data objects that have meaning in the Labanotation context. These data objects are sorted in terms of start time for a movement symbol and in terms of the column in which they are placed since this indicates the body part concerned. Modifier symbols are simultaneously collected and later associated with the symbols or columns that they modify based on their proximity. The resulting data objects are placed in a data

structure that is a conceptual analogue of the Labanotation score and is organized in much the same way as the channel data structures commonly used in animation.

This Composite Score data structure contains a map of channels (based on columns in the score), each channel being associated with a different limb, or support. The symbols within each channel are ordered by time, and each entry contains all possible modifier information. Ancillary information, such as timing, paths, or relationships is also stored in the data structure outside of the symbol channels. To complete this stage, the data object timing is adjusted to reflect the real timing of the gestures according to context dependant rules.

3.2 The LabanDancer Model

The body of the human figure to be animated in LabanDancer is based on a deformable polygonal mesh model driven by a hierarchical skeleton. The skeleton joint angles are controlled by keyframe animation channels, and additionally four inverse kinematic chains are used to drive the arms, legs, feet and hands. The inverse kinematics algorithm chosen for this purpose is IKAN (Inverse Kinematics using ANalytical solutions) [15] which is specialized for controlling human-like limbs with seven degrees of freedom and has a deterministic rather than iterative solution.

The end effectors representing foot and hand position and orientation are controlled by keyframe channels. These channels have the added ability to interpolate position either linearly or spherically, a feature that is required to create believable paths between poses.

The global position of the root of the hierarchy (i.e. the pelvis) is controlled by a three dimensional Bezier curve, the parameters of which are determined by a footsteps algorithm discussed in a later section.

There is no limit to the variety of bodies that can be animated in LabanDancer as long as they have the same kinematic model. Currently there are four separate models, two female, and two male customized for modern and classical dance.

3.3 From Composite Score to Animation

Once the composite score is derived from the LabanWriter file, the symbol channels of the resulting data structure are traversed. Each symbol type encountered for a particular body part is handled by a separate rule, which translate gestures, twists and support transitions into keyframes for limbs, end effectors or footstep position and timing information. The following sections provide some detail.

3.3.1 Gestures to Keyframes

Although Labanotation can specify arbitrary pose detail for the limbs, the bulk of the gestures are defined for the whole arm or the whole leg using the direction symbols, and optionally modifier symbols that define the extension or contraction (e.g. the amount the elbow is bent). The gesture symbols can be easily translated into spherical coordinates using a look-up table, which is used to define the spherical keyframes for the leg and arm. Some complexity is added by the fact that the coordinate frame for Labanotation gesture symbols defines up and down in global terms, but front-back and left-right in terms of the facing of the subject. The default orientation of the hands and feet are also derived from look-up tables that relate symbol type to orientation.

Symbol timing cannot be translated directly from the exact position of the symbols on the score, but must be interpreted contextually, a recurring theme in translating Labanotation. For example, although a gesture is written within a beat, the usual timing that is understood is for it to begin moving before the beat and arrive on the beat. This may vary depending on the adjacent or following symbols, for example arm gestures must be synchronized with leg gestures, which in turn are constrained by the support symbols (e.g. enough time must be left to complete the leg gesture before it contacts the ground for the next support).

Another complication related to gestures is the path the limbs must follow between gestures. In general, for arm gestures the path of the hand should follow an arc. When two sequential gestures are 180˚ apart, as when the arm travels from forward middle to back

middle, the hand will follow a straight line to the shoulder, and the elbow will bend, and then the arm will unfold as it tries to achieve the back middle position. Similarly, leg gestures normally follow a conical path with the vertex at the hip, but will follow a more complex path between extreme gestures.

In Labanotation, gestures may also be specified for sub-parts of the limbs (e.g. upper arm, lower arm, hand, fingers). This can in practice be implemented using the individual keyframe controllers for each joint in the model, but has not yet been done, since some mechanism must be designed to seamlessly blend between inverse kinematic (whole limb) and forward kinematic (limb parts) animation sources.

In general, leg and arm gestures are generated in the same way; however, leg gestures present a special problem since they must be integrated into the system that handles the support changes (i.e. locomotion). This integration is discussed in a subsequent section.

3.3.2 Support Changes to Footsteps

In Labanotation the notation shows which limb supports the body over a period of time and the direction (if any) of the movement. For example, support changes from foot to foot, combined with forward movement, result in a forward walk or run. Thus the notation does not explicitly specify the flexion and extension of the limbs concerned, leaving it to the intelligent dancer to recognize which movements are necessary to achieve support change in a specific direction. To animate this, we have to find a way to deduce the limb movements from the support changes.

We first explored an approach that was based on a stored database of animation sequences for the common transitions. Similar approaches have been implemented by Gleicher et al for the synthesis of arbitrary motion paths from motion capture data [16,17]. In many ways this worked well – the notation specialist could create a keyframe animation sequences to achieve the required transition. We implemented what was, in essence a finite state machine to formalize this. Unfortunately, however, the approach did not scale well, as the number of possible transitions rises exponentially with the complexity of movement. Labanotation support symbols only specify where to move from the current state, thus the same symbol pattern can result in completely different movement, depending on the state of the dancer before the symbols are encountered.

The analogy between support changes and footsteps is immediately obvious to all who are first exposed to Labanotation and suggests a different approach to generating animation from notation. Indeed, several techniques have been developed to reconstruct bipedal motion from the position and timing of foot placements, including that of Girard [18], and more recently, van de Panne [19]. It is this second technique, which is both simple and yields highly believable results, that we have adopted, and adapted to meet the demands of the translation process.

Van de Panne's algorithm alters the parameters of a splined curve during an optimization process that attempts to find a smooth path for the centre of mass (COM) that is constrained by a simple dynamic model and by a so-called comfort term that enforces visual plausibility. The error term that is minimized during the optimization is given by:

$$E = \int_0^T \left(\varepsilon_{physics} + \varepsilon_{comfort}\right) dt$$

$$\varepsilon_{physics} = \left\| F + mg - ma \right\|$$

$$\varepsilon_{confort} = k\left(\ell - \ell_{nom} \right)^2$$

where $\varepsilon_{physics}$ is the component of the acceleration of the mass that cannot be achieved by forces running through a simplified model of the legs (i.e. lines joining the contact points with the COM). Figures 2 and 3 show the free-body diagrams for single and double stances that illustrate the term. $\varepsilon_{comfort}$ is simply a measure of how far the leg must stretch or be compressed in order to reach the contact position.

The conversion of support symbols to footstep positions and times is fairly straightforward. Support symbols essentially indicate the direction of the next step and are positioned at some absolute time on the staff. Footsteps by

comparison, are given absolute positions, but the time is always relative to the last footfall. To determine the location of the next footfall, one simply assumes a step length appropriate for the current body model and multiplies that by the step direction vector. While that is the essence of the process, the step length is also dependant on other factors such as the direction of the step. A person normally takes a larger step forward than to the side. Likewise, the level of the step is important. Steps taken on point are smaller than steps taken on the whole foot. The step length is also increased if the subject is running or hopping, which is indicated by airtime in the score. The longer the subject is in the air, the greater the step length (to a point). Hopping produces much smaller step lengths than running. The algorithms to achieve natural steps were optimized by trial and error. The results are generally satisfactory, as can be seen from the performance of the system.

After the foot positions and times are generated, the optimizer is run to generate the COM path. Next, keyframes are generated for the feet to achieve the footstep positions at the correct times and to specify the position of the feet during the flight phase of the gait. Inverse kinematics is used to animate the other joints in the legs. The automatic generation of believable keyframes is not fully developed in van de Panne's original paper, and is one of the main contributions of our work. Figure 4 shows a schematic of the Footsteps system.

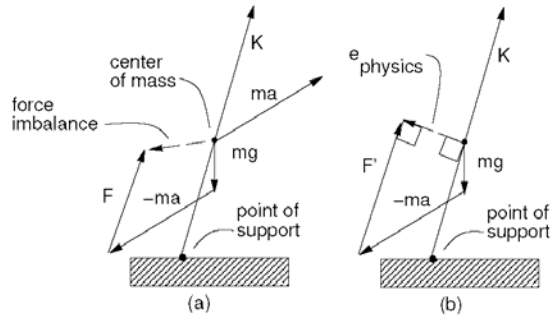

Figure 2: Single Leg Stance [19]

Foot keyframes fall into three separate categories: Ground Keys, Flight Keys and Gesture Keys. Ground keys represent the foot as it makes the transition from foot strike to foot lift-off. The position and orientation are specified in global co-ordinates. Flight keys determine how the foot behaves between foot strikes when no leg gesture has been specified. The number and position of the flight keys depends on the locomotion mode and speed. Walking, running, hopping and side stepping all require quite different keyframes. Since the mode is not explicitly expressed by either the support gestures, or the footsteps, it must be imputed from the context. Foot flight keys, positions and orientations are given in the local coordinate system of the hip. Gesture keys are derived from the gesture symbols neighbouring the support columns as described in the previous section. Gesture key positions are given in hip local spherical coordinates, while orientation keys are given in ankle local coordinates.

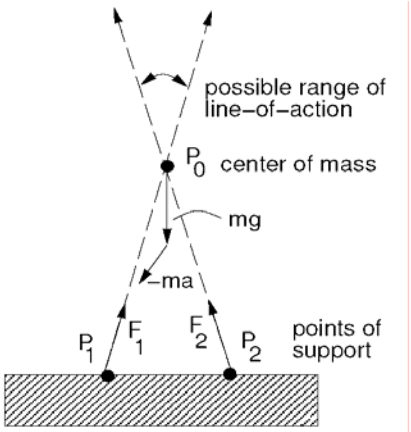

Figure 3: Double Leg Stance [19]

The interpolation engine for the foot is complicated in the sense that it must keep track of which coordinate frame it is using at any given time. In order to achieve this, the times of interpolation state changes are recorded in a separate channel.

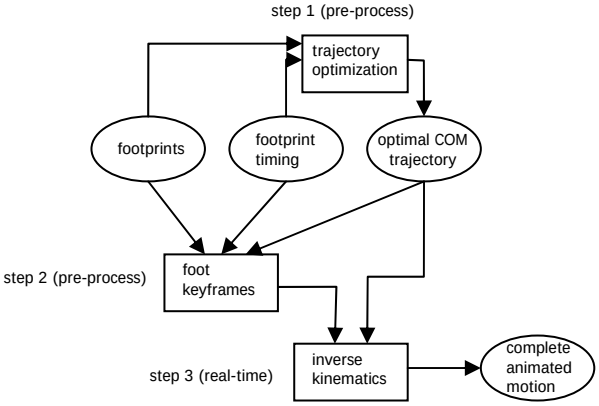

Figure 4: Footsteps system

3.3.3 Ground Keys

The time that the foot is on the ground is again divided into three distinct phases, the foot strike (heel, or toe), the foot placement, and the foot lift-off. During heel strike, the heel is constrained to stay at the same global location. The foot rotates from its strike position to the flat foot position during some fixed sub-interval of the foot contact time as illustrated in Figure 5. The instantaneous rotation of the foot about the heel contact position is simply a linear interpolation of the strike angle (red) and the flat foot angle (yellow). The instantaneous ankle position (i.e. the IK goal position) is calculated from the foot geometry.

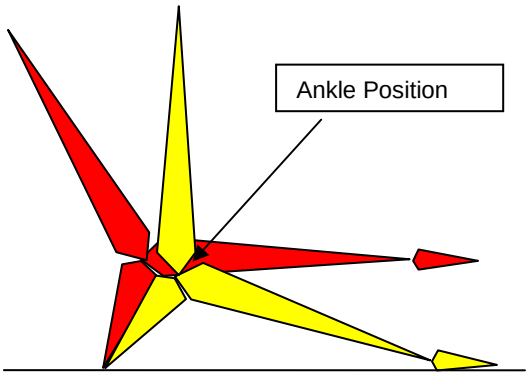

Figure 5: Heel Strike to Flat Foot

For toe strike, or toe flex when rolling off the foot, the ball of the foot is constrained to be fixed to the same global position. Additionally, the foot can be constrained to flex in a plane that is fixed with respect to the footstep orientation, or can be free to rotate about the up-axis, as when the subject pivots about the toe. The first step is to calculate the ankle position assuming that the foot is flat on the ground. If the instantaneous distance from this position to the hip is greater than the leg length, then the toe must be bent in order to effectively lengthen the reach. In this case, a new foot orientation is calculated such that the ankle toe vector is aligned with the point along the line joining the hip to the originally calculated position that is exactly one leg length away. The ankle position is then calculated from the geometry. (See Figure 6)

The toe joint is automatically controlled in such a way that it bends to avoid interpenetrating the floor. This is done by finding the intersection of the circle centred at the ball with radius equal to the toe length with the floor. The toe is then oriented to be aligned with the vector joining the ball of the foot with the correct intersection point.

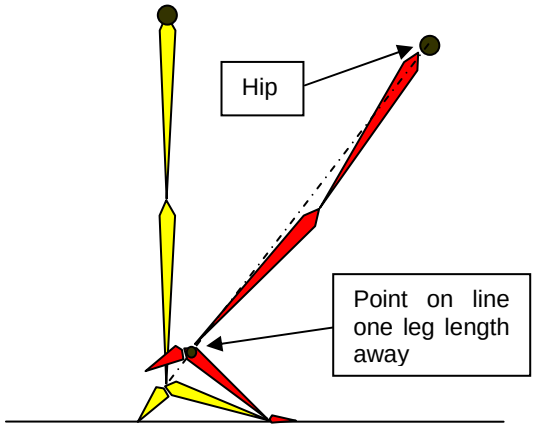

Figure 6: Foot Roll

3.3.4 Flight keys

The task of generating believable keyframes for the flight phase of the feet depends on many sometimes conflicting factors. To complicate matters, the algorithm must be able to differentiate between the different locomotion modes, only by the position and timing information provided by the footsteps.

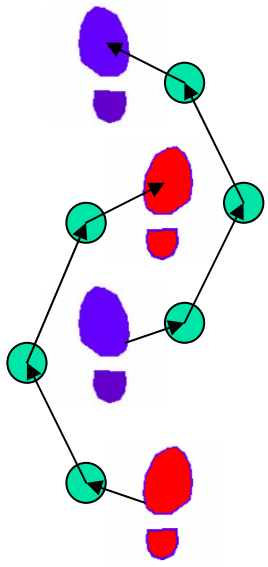

Figure 7: Extra keyframes required for crossing

In the simplest case, that of forward walking, as few as one position and one orientation keyframe are required to generate a believable path for the foot. In the case where the feet cross over, however, extra keyframes must be inserted to avoid inter-penetration of the legs (Figure 7). For running or hopping, several keys are required to generate the "kick" and the landing preparation. Timing and position are crucial to achieving a believable result, and the same keyframes may not satisfy different runs or hops, depending on speed or distance travelled. For example running and jogging require quite different flight keys. To achieve this, the base keyframes are parametrically repositioned based on time in air.

3.3.5 Leg Gestures

In general, leg gesture keys are inserted in place of the normal flight keys that would have otherwise been automatically generated. Constraints are placed on the gesture interpolation to ensure that the foot does not interpenetrate the floor. This is done by limiting the minimum altitude for the ankle joint such that it is always higher than the nominal ankle, heel distance. The ankle joint is flexed so that the ball of the foot always remains above or at the level of the floor. These measures are necessary, since the gestures are independent of the path calculated for the centre of mass by the footsteps algorithm, and otherwise the foot may be forced below the floor. In addition, special keyframes are added to the pelvis and lower back to compensate for the apparent weight imbalance when gesturing with one leg while the other leg remains fixed to the ground.

3.3.6 Holds, Turns and Pivots

A turn symbol within a support column of the Labanotation score indicates a turn or pivot. This is not taken into account in the original footsteps algorithm and thus some modifications were required. The solution we arrived at was to divide the optimization of the path into sub-segments bounded by the points where the pivoting occurs. These points are marked as fixed, and therefore not moved by the optimization algorithm. Extra keyframes are automatically inserted for the pelvis so that the model spins about its own axis by the amount specified by the symbol. The ball of the foot is constrained to be fixed to the pivot point, with the heel raised above the ground, but is allowed to swing around the vertical axis. The flight foot is held in a neutral position above the ground.

Hold symbols in the support column indicate that the subject momentarily comes to a rest. In this case, the generation of the COM path is handled in the same way as for pivots, that is, the optimization takes place across sub-segments between fixed end points.

3.3.7 Curved Paths

Labanotation uses special marginal symbols to indicate that locomotion is occurring along a curved path. Each curve symbol normally indicates the angle of the turn as some fraction of a full circle. The support changes across which the curving takes place are indicated by the span of the symbol. The amount of circling is divided amongst the support symbols by the translator.

3.4 The User Interface

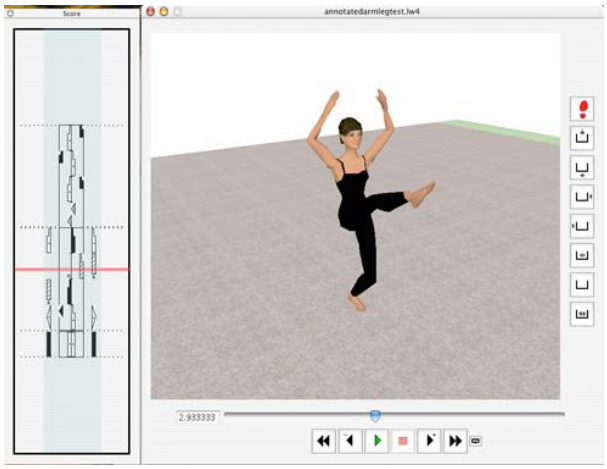

Figure 8: The User Interface

LabanDancer is equipped with a straight-forward user interface as illustrated in Figure 8. The single window is composed of a rendered sub-window showing a dancer on a stage, buttons to choose one of the cardinal camera views and a standard set of playback controls. The rendered sub-window responds to mouse events allowing the user to adjust the viewpoint by orbiting, panning or zooming the virtual camera. Different dancer models are selectable from a drop down menu, as are

options to turn on or off an audible metronome, or change the preference from heel-strike to toe-strike locomotion. The open dialog to choose a LabanWriter file is invoked from the File menu. The placement of the calculated footstep positions can be optionally displayed on the stage.

An important feature of the interface is the display of a simultaneous graphic of the original Labanotation score on the left of the screen. A cursor moves up the score as the animation progresses, and can be "dragged" to set the desired time while the playback is stopped.

4. Future Work

LabanDancer is a useful working prototype – although incomplete it can already serve as a teaching tool. A major research focus for the future centres around the idea of deriving a digital representation of Labanotation that is independent of how the Labanotation score is drawn on the page. Current Labanotation editors produce files that store the elements of a dance score as graphical objects with x and y coordinates. The problem with the strictly graphical approach is that, while it is sufficient for viewing or printing scores, the score itself does not lend itself to movement analysis. Our current plan is to define an XML schema to represent Labanotation. We believe that such an interchange format could have much broader application than just creating and disseminating dance scores, and could become a standard way of representing human motion, which until now, has been absent.

Another issue is the need to automatically produce believable human motion from symbolic representations. While LabanDancer does a passable job at producing motion for those parts of the body for which we have explicit notation, movement in the rest of the body is largely ignored, or treated in an ad-hoc way, such as rotating the pelvis and back joints to compensate for leg gestures. A more sophisticated model for the human body and how it moves is required to produce natural looking human motion. Baerlocher [20] has produced a number of IK techniques that can have multiple secondary goals such as maintaining balance, focus of attention, or orientation of limbs. While these techniques are not yet real time, similar general purpose approaches need to be found.

The current incarnation of LabanDancer deals only with symbols that explicitly specify either gestures or support changes. Labanotation is rich with symbols that either convey contextually relevant information such as spatial relationships between body parts, or stylistic notions such as effort or grace. Likewise, Labanotation can describe the interaction between multiple dancers, specifying touch, and proximity.

In the immediate future, we plan to implement a system that can blend animation from multiple sources. In this way, LabanDancer will be able to take advantage of a wider variety of animation techniques simultaneously in order to produce more exactly the desired result.

5. Conclusion

One of the main contributions of this work is to demonstrate the feasibility of using a symbol based representation as a compact, yet sufficient interchange format for human motion. We believe that in the 76 years since its invention, Labanotation has developed into a richly descriptive language, and a digital version would be ideally suited as such an interchange format. It would also serve to link currently disparate fields such as choreography, dance notation, human animation and motion analysis.

Other contributions of our work have been in the development of procedural techniques for the generation of human motion. This has wider application in such areas as computer games [21].

6. Acknowledgements

The authors acknowledge support for this project provided by a grant to the Dance Notation Bureau from the US National Endowment for the Humanities. Earlier support was provided by the US National Endowment for the Arts and the National Initiative to Preserve America's Dance (NIPAD). We also acknowledge Dr. Michiel van de Panne for advice on the footprints algorithm and David Eberly for the use of the WildMagic 3D graphics API.

7. References

[1] Noll, A. M. Choreography and Computers. Dance Magazine, pp. 43--45, January 1967.

[2] Cunningham, Merce, Changes/Notes on Choreography, F. Starr, ed., Something Else Press, New York, 1968.

[3] Wolofsky, Zella, Computer Interpretation of Selected Labanotation Commands. M.Sc. thesis, Simon Fraser University, 1974.

[4] Badler, N. and Smoliar S. Digital Representations of Human Movement. Computing Surveys, 11(1):19--38, March 1979.

[5] Brown, M., Smoliar, S., and Weber L., Preparing Dance Notation Scores with a Computer. Computers and Graphics, 3(1):1-7, 1978.

[6] Calvert T.W. and Chapman J., Computer Assisted Notation of Human Movement, in Proc. of 1978 ACM Conf., pp. 731-736, 1978

[7] Venable L., Sutherland S., Ross L., and Tinsley M. LabanWriter 2.0, Ohio State University, 1989.

[8] Calvert T.W., Welman C., Gaudet S., Schiphorst T. and Lee C.. Composition of Multiple Figure Sequences for Dance and Animation. The Visual Computer, vol. 7, pp. 114-121, 1991.

[9] Ryman, R. and Hughes-Ryman R.. The MacBenesh Editor: A Word Processor for Benesh Notation. Dance Notation Journal, 4(2):16-26, Fall 1986.

[10] Gray J.. Dance in computer technology: a survey of applications and capabilities. Interchange, 14(4):15-25, Winter 1984.

[11]] Hunt, F.E.S., Politis G. and Herbison-Evans, D., LED & LINTER: An X-WINDOWS Mini-Editor and Interpreter for LABANOTATION, Technical Report 343, Basser Department of Computer Science, University of Sydney, April 2003.

[12] University of Birmingham: Calaban Project. http://www.bham.ac.uk/calaban/frame.htm (accessed July 2005).

[13] Calvert, T., Wilke, L., Ryman, R. and Fox, I., .Applications of Computers to Dance, IEEE Computer Graphics and Applications, Vol: 25, No: 2, pp. 6-12, March-April 2005.

[14] Hutchinson-Guest, A., Labanotation: The System of Analyzing and Recording Movement, 3rd Edition, Theatre Arts, 1987.

[15] Tolani, D., Goswami, A.and Badler, N., Real-time inverse kinematics techniques for anthropomorphic limbs, Graphical Models, vol. 62, pp. 353-388, 2000

[16] Gleicher, Michael, Shin, Hyun Joon, Kovar, Lucas and Jepsen, Andrew. Snap Together Motion: Assembling Run-Time Animation,. 2003 Symposium on Interactive 3D Graphics. April 2003.

[17] Kovar, Lucas, Gleicher, Michael and Pighin, Frederic. Motion Graphs. ACM Transactions on Graphics, 21(3), pp. 473-482 (Proceedings of SIGGRAPH 2002). July 2002.

[18] Girard, M., "Interactive Design of 3D Computer-Animated Legged Animal Motion". IEEE Computer Graphics and Applications, Vol. 7, No 6:39-51, 1987.

[19] van de Panne, M. From Footprints to Animation. Computer Graphics Forum, Volume 16, Number 4, pp. 211-223, October 1997.

[20] Baerlocher, P., and Boulic, R. An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels, The Visual Computer, Springer Verlag, 20(6), 2004

[21] Mizuguchi, Mark, Buchanan, John and Calvert, Tom, Data driven motion transitions for interactive games, *Proc. Eurographics 2001*, Manchester, UK.