

Applications of Computers to Dance

Tom Calvert
*Simon Fraser
University*

Lars Wilke
*Credo
Interactive*

Rhonda Ryman
*University of
Waterloo*

Ilene Fox
*Dance Notation
Bureau*

Of all the art forms, dance has probably been the slowest to adopt technology. In part, this reflects the reluctance of dancers and choreographers to let anything get between them and the live kinesthetic experience. But it also reflects the fact that useful tools have been slow to develop in a market with limited commercial opportunity.

Probably the first article suggesting the application of computers to dance was published by A. Michael Noll in *Dance Magazine* in January 1967, although New York choreographer Merce Cunningham discussed this at about the same time.¹ These initial discussions suggested visualizing dance with animated stick figures and using these visualizations to plan choreography. Soon other choreographers suggested using computers to compose and edit dance notation scores and then to translate these scores into animation. Dance notation is analogous to music notation, but is less widely known—the “Labanotation Primer” sidebar provides a primer on the notation system most widely used in North America.

As computer technology has developed and become less expensive, many artists have found ways to use it to enhance their performances with interactive multimedia. This has included incorporating computer-generated images and sound with live dance performance and using sensors that let the live dancers’ movements control imagery, sound, and a wide variety of special effects.

This overview describes some of the current applications of computer graphics to dance including visualizing choreography, composing, editing and animating dance notation, and enhancing live performance.

Visualizing dance and planning choreography

When composing a dance, a choreographer typically works with one or more dancers to physically set the movement of each dancer over time. For any significant piece, this can take weeks or months. The choreography requires not only the dancers’ time but also access to appropriate rehearsal space in a dance studio. This is expensive and dance companies work on restricted budgets. A computer tool that could be used to plan the choreography would help minimize the time needed for live rehearsal.

In 1991, the founders of Credo Interactive—the developer of Life Forms, a software package for planning

human movement—began to work with Cunningham and other choreographers to explore this approach. The result was the evolution of Life Forms into a tool known as Life Forms Dance and now as DanceForms (<http://www.danceforms.com>). This tool lets the choreographer try out ideas before ever meeting with live dancers. None of DanceForms’ features would surprise those working in human figure animation; however, it’s customized for dance. Similar features exist in Curious Labs’ Poser, Alias’ Motion Builder and in general-purpose animation systems such as Alias’ Maya and Discreet’s 3ds max. However, most of those applications are opaque to the typical dance choreographer or dance teacher.

As Figure 1 (page 8) shows, DanceForms provides the choreographer with a stage, where figures can be placed. While it’s possible to use any body with an articulated skeleton (for example, a spider, a horse, and so on), the standard bodies provided are male and female human figures for ballet and modern dance. After selecting a figure or figures, a keyframe animation approach builds up the movement for each of the dancers. A studio window provides a flexible set of controls to refine the posture of individual figures, including 3D potentiometers for joint angles. Limbs and limb segments can be directly manipulated with the mouse, with or without inverse kinematics (the IK algorithm calculates joint angles of a limb as the user positions the endpoint). It’s also possible to import motion capture data using one of the standard formats.

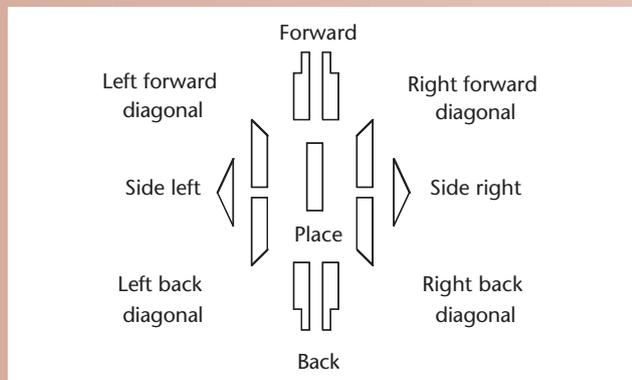
As the user adds keyframes, their sequence can be viewed and edited in the score window, which is essentially a timeline. The partially composed piece can be animated at any time in the stage window, using a variety of body representations including stick figures, bounding boxes, outlines, or mesh surfaces. Alternatively, the piece can be rendered without perceptible delay and played in the 3D performance window.

A wide variety of choreographers—including Cunningham and Danny Lewis—have used this choreography package. Educators in K–12 and postsecondary institutions also use it—particularly innovative educational users include the states of Kentucky in the US and of New South Wales in Australia. While the tool has merit in its own right as an approach to thinking about movement in 3D and across time, it helps with resources by letting students in more common computer labs plan

Labanotation Primer

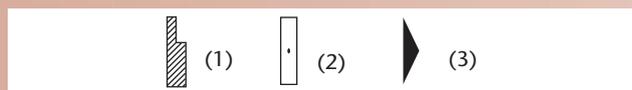
Labanotation is a system for analyzing and recording human movement. The original inventor, Rudolf von Laban (1879–1958) was an important figure in European modern dance. He published this notation first in 1928 as *Kinetographie*. Several people continued the development of the notation and Hutchinson Guest has written the classic text.¹ In the US, the New York-based Dance Notation Bureau supports Labanotation and archives many Labanotation scores (<http://www.dancenotation.org>).

Each Labanotation symbol gives four pieces of information. First, the symbol's shape indicates the direction of movement (see Figure A).



A Direction symbols.

Next, the symbol's shading shows the level of a movement; diagonal strokes for high, a dot for middle, and blackened for low (see Figure B).



B Symbol shading: (1) forward high, (2) place middle, and (3) right side low.

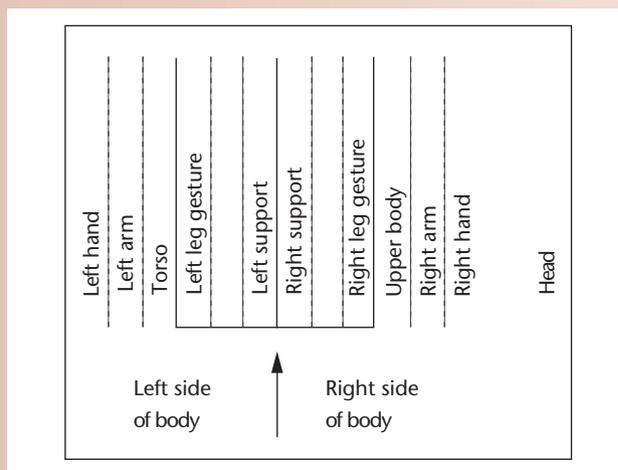
Third, the symbol's placement on the column on the staff indicates the part of the body that is moving. A Labanotation staff represents the human body; the center

line divides the left side of the body from the right. Symbols to the left of the center line refer to the left-hand side of the body, symbols to the right of the center line to the right-hand side of the body. Some body parts must be identified by a symbol, see Figure C.



C Symbols indicating body parts: (1) head, (2) face, (3) hands, and (4) front of left shoulder.

Finally, the symbol's length indicates duration of the movement. The staff is read from the bottom up; moving ahead in time (see Figure D). The tick marks on the center line divide the time into counts and the horizontal lines correspond with the bar lines in the music. Movements written on the same horizontal line occur simultaneously; movements written one above another occur sequentially. Measure numbers and dancers' counts appear to the left of the staff.



D The staff.

Reference

1. A. Hutchinson Guest, *Labanotation: The System of Analyzing and Recording Movement*, Taylor and Francis, 1987.

choreography that is later performed in scarce dance studios. They can then make better use of the time spent with the dancers.

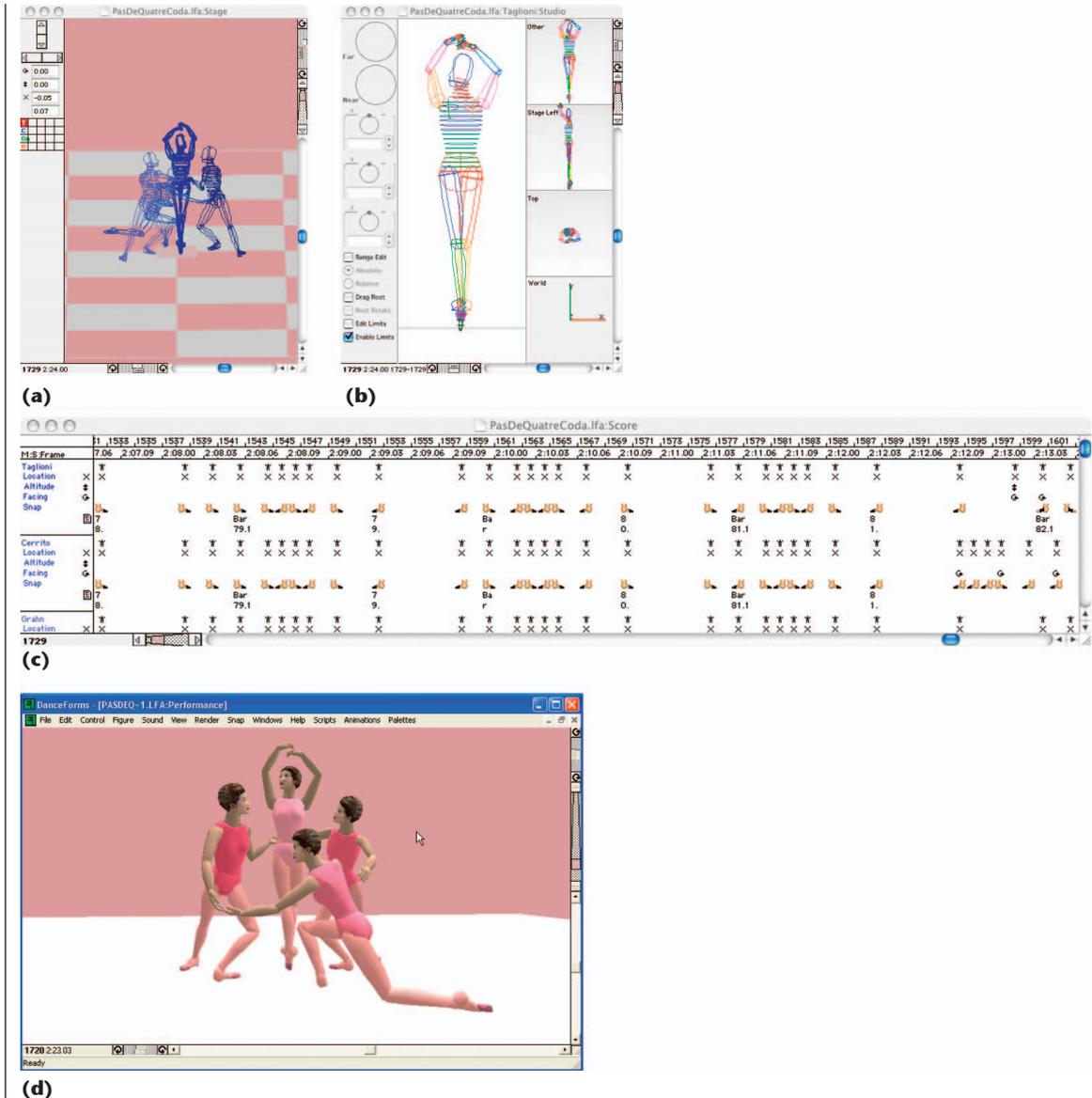
DanceForms is suited as a tool in dance education, introducing students to a variety of dance movements and allowing them to explore new ways to expressively combine those movements. Rhonda Ryman at the University of Waterloo has created comprehensive content including dictionaries for ballet (*Ballet Moves II*) and modern dance (*Modern Dance Moves*); both are published by Credo Interactive (<http://www.danceforms.com>). *Ballet Moves II* details hundreds of positions and steps as codified in the English and Russian schools of ballet, as well as select samples from the Italian school. Figure 2 shows an example. This con-

tent provides the teacher and the student with a rich environment in which to explore dance animation. Complementing these modern dance and ballet moves content collections, Ryman has published a tutorial-based practical guide to *Dance Forms 1.0* (<http://www.dcd.ca>) offering a hands-on approach to learning the program with step-by-step exercises.

Composing and editing dance scores

The Labanotation primer in the sidebar shows that a score consists of symbols on a vertical staff. The symbol's meaning depends on its shape and shading, vertical length, and placement in one of the staff columns. Several word-processing-type editors support the composition and editing of Labanotation scores. These include

1 DanceForms provides (a) a stage window for composing multiple dancers, (b) a studio window for creating particular body positions, (c) a score window to show how each dancer moves over time, and (d) a rendered performance window.



- AutoCAD-based Calaban, which is particularly useful for creating high-quality printed scores (<http://www.bham.ac.uk/calaban/frame.htm>);
- AutoCAD-based Labanatory, which aims to let users search a score (<http://www.labanatory.com/>);
- X-Windows-based Labanotation LED developed in Australia (<http://www-staff.it.uts.edu.au/~don/pubs/led.html>);
- MacBenesh for Benesh notation, developed by Ryman and her colleagues at the University of Waterloo and now maintained and distributed by DanceWrite (<http://members.rogers.com/dancewrite/index.html>); and
- Benesh Notation Editor developed by the Benesh Institute in London for editing multistave notation scores (<http://www.benesh.org/frames.html>).

These editors work well, facilitating the writing and more importantly the editing of scores. However, the most widely used editor is the LabanWriter (<http://www.dance.ohio-state.edu/labanwriter>). This

Macintosh-based 2D graphics editor specializes in creating Labanotation scores (a Windows version is being developed). It was developed at the Dance Department of the Ohio State University by George Karl, Scott Sutherland, and David Ralley under the direction of Lucy Venable. LabanWriter lets the user create one or more staves on the virtual page and provides a palette of Labanotation symbols that can be selected and placed on the staff in the columns representing the appropriate body part. Direction and turn symbols can be stretched in the vertical direction to indicate their duration. Modifier symbols can be selected and placed in proximity to the main symbols or columns that they modify. The development of a score is illustrated in Figure 3.

LabanWriter treats symbols strictly as 2D graphical objects that fall loosely into two subclasses: stretchable and fixed sized. Stretchable symbols have associated time durations, such as direction symbols and turn symbols, while fixed size symbols are normally modifiers of columns, measures, or other symbols, or are used to annotate the score.

Column and timing information is not explicitly stored with the symbols. The symbols are organized into columns and measures only insofar as their Cartesian positions indicate. LabanWriter has no knowledge base for parsing the score and nothing prevents the user from entering complete nonsense.

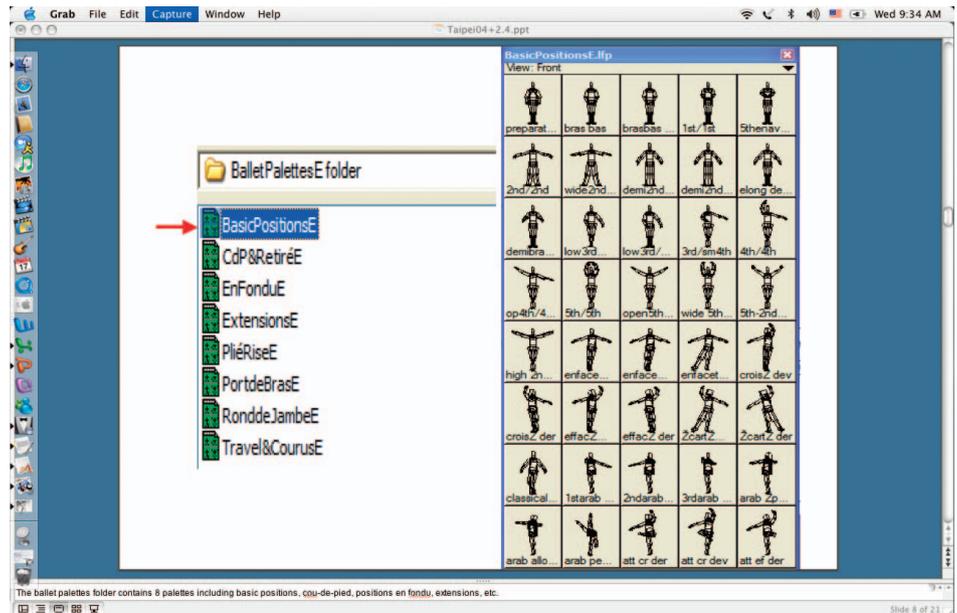
The LabanWriter file (.lw4) is an ASCII file that stores only enough information to reproduce the arrangement of the graphical symbols just described. Each record in the file contains the type, position, size, color, special fonts, and other information related to the drawing of the symbols. The order of symbols within the data stream is the one laid down by the user. No attempt is made to sort the symbols in any other way. (A detailed description of the file format is available at <http://www.dance.ohio-state.edu/labawriter/LW4/LW44FileFormat.html>). LabanWriter can also print the Labanotation score, or produce a 2D raster image in one of three popular graphic file formats (.png, .pict, or .jpeg).

LabanReader, also developed at the Ohio State University, is a software program for selective viewing of notation symbols in a dance score. It can be used to focus on particular aspects of movement for teaching or to display various kinds of patterning for movement analysis. LabanReader is suited for use with scores created in LabanWriter 4.0 or later.

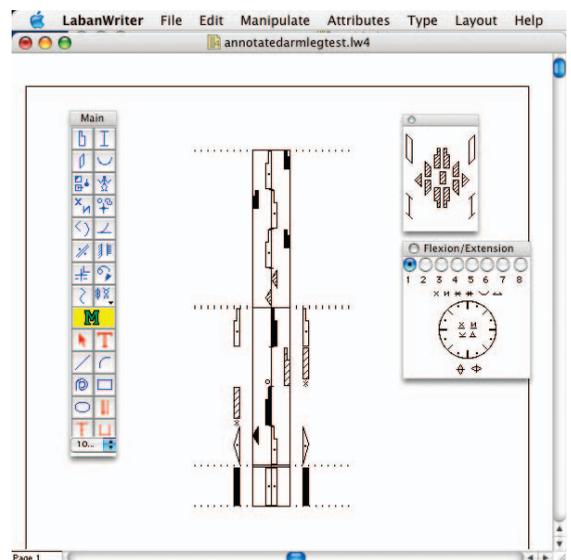
As mentioned previously, LabanWriter and other notation editors fulfill a useful function. However, to date, the developers have not accounted for knowledge of the score's structure (syntax or semantics). In contrast, almost all word processors check spelling and many make suggestions on grammar. The notation editors so far have not applied rules to check the structure of what has been composed—they are merely drawing systems for placing symbols on a staff. Another issue is their inability to search a score for a particular movement pattern (Labanatory has some searching capabilities). One reason that notation editors are so limited is that no consistent, unambiguous way exists of representing human movement with a machine-readable ontology or grammar of the kind that can be created in XML, for example.

Translating dance notation into animation

Not everyone in the dance community can read notation, and writing dance scores is a specialized skill. A program to translate notation into animation (and vice versa) could make the notation more accessible to choreographers, dancers, and students. The New York-based Dance Notation Bureau, in collaboration with Simon Fraser University, University of Waterloo, and



2 Ballet Moves II provides a dictionary of ballet positions and movements, including this palette of basic positions in the English school.



3 Composing a score with LabanWriter. The user selects from the main menu (above left) to see pop-up menus containing families of symbols or signs (above right) that can be placed on the three-lined vertical staff.

Credo Interactive, has led the development of LabanDancer, a program to translate LabanWriter scores into 3D animation.

Creating a composite score

The first stage in translating LabanWriter files is to convert the stream of graphical symbols into a set of composite data records that have meaning in the Labanotation context and to populate a data structure that is a more direct analog of the Labanotation score. In this process, graphical symbols are spatially sorted and assigned to columns and measures based on the

symbol position relative to the staff origin. Modifier symbols are simultaneously collected and later associated with the symbols or columns that they modify based on their proximity (an inherently error-prone operation).

The composite score thus derived contains a map of animation channels; each channel is associated with a different limb segment or support (in most cases this corresponds to a column in the Labanotation score). The data records within each channel are ordered by time, and each record contains all possible modifier information. Ancillary information, such as timing, paths, or relationships is also stored in the data structure outside of the channels. In a final pass, the data record timing is adjusted to reflect the real timing of the gestures according to context-dependent rules.

Kinematic model

The LabanDancer application uses a deformable, polygonal mesh model driven by a hierarchical skeleton. Keyframe animation channels control joint angles in the skeleton, and four inverse kinematic chains drive the arms, legs, feet, and hands. The software uses the IK using analytical methods (IKAN) algorithm, which is specialized for controlling human-like limbs with 7 degrees of freedom and has a deterministic rather than iterative solution.² IKAN also has the advantage of allowing control over the elbow or knee joints, which is important for achieving all possible poses specified by the Labanotation score. Keyframe channels control the end effectors, representing foot and hand position and orientation. These channels can interpolate position either linearly, or spherically, a necessary feature in creating believable paths between poses. A 3D Bezier curve—the parameters of which a footsteps algorithm determines—controls the global position of the root of the hierarchy (that is, the pelvis).

Currently, LabanDancer provides four separate models, two female and two male, customized for modern and classical dance. Users can easily create more models representing different dance traditions by using commercially available modeling programs.

Three command types

The composite score is parsed into three streams. The first interprets gestures—these are non-weight-bearing movements of any body part. The second interprets support changes (including locomotion), and the third involves other issues such as repetition of a sequence, paths, use of floor plans, and so on.

While much of Labanotation is explicit—it objectively specifies the orientation of limb segments at a particular time—there are numerous instances where the Labanotation is implicit—relying on notator and dancer knowledge to deduce the movement in the context of the piece in question. Thus, the translator program must include a knowledge base from which movements can be interpreted in their context.

Interpreting gestures

The Labanotation symbols generally indicate unambiguously the start time, end time, and final orientation of limb parts involved in a gesture. They do not, howev-

er, explicitly specify the path to be followed in carrying out the gesture. The path can be deduced once the starting and ending orientations are known. LabanDancer uses an inverse kinematics algorithm and applies constraints to ensure that articulated limbs carry out the movement in an appropriate plane. For some movements it's necessary to add intermediate keyframes as additional constraints to ensure that articulated limbs do not move inappropriately, such as passing through other body parts, for example.

The parser that interprets gestural commands is simple at a high level—as noted, final limb orientations are usually explicitly specified. However, at a detailed level the parser is extremely complex since it must ensure that every movement is appropriate to the context. In addition to the constraints discussed previously, other contextual issues include the dance style—for example, a foot might be pointed in ballet, but not in folk dance—and the timing conventions used in Labanotation—although a gesture is written within a beat, the usually understood timing is for it to begin moving before the beat and arrive on the beat. This can vary depending on the context. For example, arm gestures must be synchronized with leg gestures, which in turn are constrained by the support symbols—enough time must be left to complete the leg gesture before it contacts the ground for the next support, and a leg cannot gesture until weight is transferred to the other leg partway through the step.

Interpreting support changes

In Labanotation the concept of support and support change is the basis for all locomotion. The notation shows which limb supports the body over a period of time and the direction (if any) of the movement. Support changes from foot to foot combined with forward movement, for example, 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.

One approach to the animation of support changes would be to store keyframe sequences for all possible transitions; however, the number of sequences increases exponentially with the movement's complexity. LabanDancer uses a general approach based on van de Panne's footprints algorithm for animation of human locomotion.³ This algorithm calculates an optimum path for the center of gravity of a biped based on the placement and timing of the footsteps and the geometrical configuration. Once the path of the center of mass is known, keyframes for the feet are generated, based on the position and timing of footsteps and any restrictions imposed on the flight phases of the foot. The foot keyframes drive the goal positions for the IK chains driving the legs. Important in the choice of this algorithm was the fact that it could take changes in level into account, something that is important in Labanotation. Also, the algorithm seamlessly handles the transition from walking to running, jumping, or hopping, and can handle locomotion along a curved path.

To implement this approach it's necessary to find a

way to generate the timing and position of the footprints from the composite Labanotation score. The footprints algorithm specifies the footsteps in absolute position and relative (to the last) time, while Labanotation specifies support changes in absolute time and relative (to place) position. The translation from one system to the other is relatively straightforward.

Turns, pivots, and other commands

Turn symbols within the support columns of the Labanotation score indicate a turn or pivot. The footsteps algorithm that we use does not adequately take this type of motion into account, and thus some modifications were required. Also, many other Labanotation symbols do not translate into either gestures or changes in support. These include techniques to handle multiple dancers, the use of floor plans, symbols to indicate repetition of a sequence, and so on. The parser recognizes these symbols and has procedures to implement the corresponding movement.

LabanDancer prototype

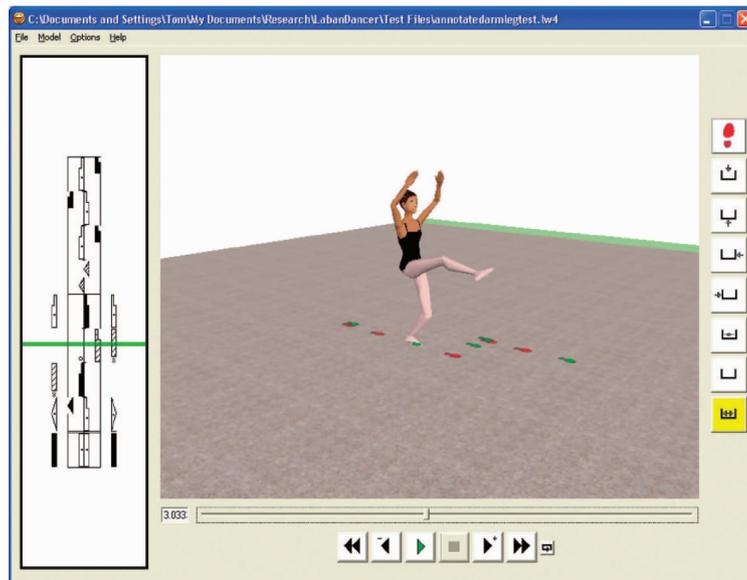
The first LabanDancer prototype has been implemented for Windows and Mac OSX. Figure 4 shows the user interface. Fairly standard controls allow the user to adjust the viewpoint by rotating the stage on all three axes and by providing for zooming in and out. In addition to totally interactive control with the mouse, fixed views can be chosen or the camera can follow the dancer. The display can show the dancer's footprints, defining the path of the movement. Also, a sound output provides a metronome-like tock sound to indicate each beat. An important interface feature 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. The user can drag the cursor on the score or on the timeline below the animation to move backwards and forwards through the animation sequence.

Future

This human animation software currently translates numerous types of movement including supports, gestures, turns, circular paths, and contracted limbs. Future versions will be extended to handle all aspects of a score and in addition, LabanDancer will be written into a file that can be read into DanceForms or other more general programs for human figure animation. This file can then be edited and combined in a variety of ways with other sequences from animation, motion capture, or LabanDancer translations of other Labanotation scores. A more complete description of LabanDancer can be found elsewhere.⁴

Current research and future developments

There are many things that can be done to enhance and extend the computer technology used to animate and visualize dance, plan choreography, edit and animate notation, and enhance performance. However, a recurring issue in many of these developments is the need for a unique, unambiguous way to represent



4 LabanDancer interface lets the user follow the animation of a Labanotation score. Using the score or animation cursor the user can go back and forth, comparing the movement with the notation.

human movement. Notation systems, such as Labanotation, do this at a conceptual level, but since the notation itself does not have an unambiguous machine-readable representation, the notation is not enough. The composite score data structure developed for LabanDancer is a pragmatic attempt to address this problem. A more principled approach must start by defining an ontology or grammar that captures the necessary syntax and semantics.

In April 2004, the Dance Notation Bureau and the Ohio State University sponsored a workshop where 29 experts in Labanotation, dance, human movement, and technology met to discuss adding intelligence to programs that compose, edit, and animate Labanotation. This would add understanding of a symbol's attributes, such as which body part is moving, timing, and so on, and enable new advances in the use of technology and notation. (See the meeting report at <http://dancenotation.org/DNB/news/report.html>.) The consensus at the workshop was that there is a need for an interlingua—that is, an unambiguous representation of movement that can map to different notation systems and editors, animations, and motion-captured dance data.

Like all efforts to develop international standards, this is a challenging problem. Following the development of MusicXML to provide structure for musical scores, there has been some initial discussion of a possible LabanXML as proposed by Minako Nakamura and her colleagues in Japan⁵ or a more comprehensive DanceXML as proposed by Simon Fraser University graduate student Jonathan Hatol. It's easy to see how XML can help capture the high-level structure of Labanotation, but it's more difficult to capture the more detailed context-dependent structure.

When a usable ontology-based representation is developed, it will then be necessary to create interfaces to the existing representations used in notation editors in sys-

Computer Graphics and Interactive Technologies in Live Performance

There are many examples of the use of all forms of multimedia in conjunction with live dance performances. In some cases a preplanned animation sequence or digital video with appropriate sound is played as a backdrop and live dancers interact with the preprogrammed display. A more technically challenging system is required when it senses the live dancers' movements and modifies the imagery and/or sound. Another situation arises when digital video, possibly with immersive or stereo display, links two distant locations for rehearsal and/or performance. All kinds of other variation are possible.

One ambitious example is the *Trajets* (<http://www.trajets.net/>) interactive installation codirected by Susan Kozel and Gretchen Schiller, with development by Rob Lovell and Scott Wilson. Here, 12 trapezoid-shaped, motorized screens are

suspended from a grid and spiral in response to visitors. Vision-sensing-software eyes let the computers localize the visitors' positions in space. Different visual sequences are projected on each of the screens. The visuals, which experiment with body movement, have a unique digital look and feel that the artists created in the video shooting process. One image captures a person slowly jumping and falling and other images show bodies in mud and water—Figure A gives a sense of this complex environment.

Another exploratory project is *Whispers* (<http://whisper.surrey.sfu.ca/index2.html>) by Thecla Schiphorst and others, which involves physiological sensors, small wearable computing devices, wireless computer communication, and handheld technologies embedded in evocative and playful garments worn by the participants. It maps physiological data onto linked and networked devices worn on or close to the skin and in garments. In other words, it involves collecting data from the bodies of participants and communicating the participants' responses to the system and other participants through visualization and sonification techniques.

One venue where much of the latest work is displayed is the biannual Monaco Dance Forum (<http://www.monacodanceforum.com>). The forum is an international dance meeting and festival that offers dance performances, multimedia installations, exhibitions, video dance screenings, and symposiums.

Another source for work up to 2000 is the Digital Performance Archive (http://dpa.ntu.ac.uk/dpa_site/).



A View of the *Trajets* interactive installation.

Courtesy Roth-Kimura Architects, Honolulu, HI

tems for animating dance and in motion capture systems. The great advantage of this new representation will be that it will define all movement explicitly and not implicitly, as with present systems. It will be possible to search easily for movement patterns and to use this power in choreography, as well as many aspects of dance scholarship. This development can also be useful in other applications that require the analysis of human movement such as animation for games or the representation of motion-captured human movement data used in the clinical diagnosis of locomotor problems. ■

References

1. M. Cunningham, *Changes/Notes on Choreography*, F. Starr, ed., Something Else Press, 1968.
2. D. Tolani, A. Goswami, and N. Badler, "Real-Time Inverse

Kinematics Techniques for Anthropomorphic Limbs," *Graphical Models*, vol. 62, no. 5, 2000, pp. 353-388; <http://hms.upenn.edu/software/ik/ik.html>.

3. M. Van de Panne, "From Footprints to Animation," *Computer Graphics Forum*, vol. 16, no. 4, 1997, pp. 211-223.
4. L. Wilke et al., "Animating the Dance Archives," *Proc. 4th Int'l Symp. Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)*, Eurographics Assoc., 2003, pp. 91-99.
5. M. Nakamura, "Text Representation of Labanotation Data for Computer Based Motion Analysis," presented at the *World Dance Assoc./Int'l Council of Kinetography Laban/Congress on Research in Dance Int'l Conf.*, 2004; http://www.imb.is.ritsumeai.ac.jp/~hachi/hachi_e.html.

Readers may contact Tom Calvert at tom@sfu.ca.

Readers may contact Mike Potel at potel@wildcrest.com.