Can Computers Learn to Dance?

Elizabeth Bradley, David Capps, and Andee Rubin

Department of Computer Science
Department of Theater and Dance
University of Colorado
Boulder, Colorado 80309 USA

`[elizabeth.bradley,david.capps, andee.rubin]@colorado.edu`

Web: `http://www.cs.colorado.edu/~lizb/chaotic-dance.html`

Published in *Proceedings International Dance and Technology (IDAT-99)*, Tempe AZ; February 1999

1 Introduction

In the past few years, the dance community has begun to make extensive use of computer animation software in representation and rendering tasks. Computer *hardware* has also seen increasing use, primarily as a means to augment and/or amplify performances in various interesting ways — e.g., a set of sensors on a performer's body, connected through a data-processing channel to a synthesizer or a lighting setup. In both of these applications, however, the computer is simply an external aid to human creativity; it is not an active participant in the work. We are interested in a wholly different type of computer tool, one that plays a truly *active* role in both the creation and the analysis of original dance sequences. In this paper, we describe several implemented programs that use cutting-edge computer-science techniques to do exactly that.

Working from examples — a single animated movement sequence or a *corpus* containing many such sequences — two of these programs automatically generate innovative and yet stylistically consonant sequences. The first of these two, called CHAOGRAPHER, uses the mathematics of chaos to operate as a "shuffler" of movement phraseology in a manner akin to certain postmodern choreographic strategies. The second program, MOTIONMIND, uses machine learning algorithms to capture the stylistic rules implicit in a given body of dance phrases. MOTIONMIND then uses that knowledge to create completely original movement sequences that retain the stylistic stamp of the given material. Both of these *artificial intelligence* programs truly get the computer "inside" the dance — unlike rendering with animation software like Life Forms, which does not "teach" a computer to dance any more than entering a text file in a word processor "teaches" that computer how to understand the corresponding paragraph of *War and Peace*.

Finally, computers can also play a more-active role in the analysis of the spatial aspects of movement. We briefly describe a class of computer programs called *Video-Based Labs* (VBLs) which can effectively augment established methods of movement and style analysis. VBLs have a number of potential pedagogical applications in the areas of choreography, technique, and kinesiology, and we are beginning to explore some of these applications.

Website

In order to demonstrate the capabilities of these programs, we have set up a comprehensive website[1], listed at the beginning of this paper, which includes universal access to:

- the CHAOGRAPHER code for generating chaotic variations

[1] Full functionality requires Netscape Navigator 4.05 or higher.

- a simple movement animation package akin to Life Forms but written in Java, and
- an interactive library of movement animation sequences, including a variety of examples of CHAOGRAPHER and MOTIONMIND results.

We hope that this interactive repository of computer tools and movement sequences will become a major resource for future experiments with both analytical and creative projects in the dance community.

2 Using Chaos to Create Choreographic Variations

Artificial intelligence[12] (AI) is the branch of computer science that strives to automate perception, reasoning, and action. It is, of course, impossible to automate the creative process, but AI can provide some provocative suggestions. As an instance, we describe the CHAOGRAPHER program[3, 4], which uses the mathematics of chaos to generate innovative variations on a pre-defined movement sequence.

Though its name suggests randomness, chaos is actually quite orderly. Chaotic systems have a fixed, fundamental structure; at the same time, they are exquisitely sensitive and react strongly to stimuli — all the time remaining within the limits of their fixed structure. Consider an eddy in a stream. Its roils and ripples are highly complex, but its general shape, size, position, etc. are constant. Two wood chips dropped a short distance apart will take vastly different paths through the eddy, but both will cover the same complicated patch of currents and cross-currents.

CHAOGRAPHER maps a pre-defined dance onto this sensitive structure and then generates new dances by dropping new wood chips and following their paths. Imagine a camera suspended above the eddy, shooting ten frames a second. This sequence of photographs is a map of the original path of the chip. CHAOGRAPHER links this chaotic map to a predefined movement sequence as follows. It first uses computational geometry[6] techniques to identify and define the region around the wood chip in each frame. It then labels the first of these regions with the first dance posture, the second region with the second posture, and so on. Imagine, now, a computer running Life Forms, hooked up to the camera suspended above the eddy, and programmed to recognize what region the wood chip is in[2] and display the corresponding body posture. If one dropped a second wood chip into the eddy at *exactly* the same starting position as the original one, it would follow the same path, and so the playback device would re-create the original sequence. If the second chip is dropped into a different part of the eddy, however, it will — as described at the end of the previous paragraph — follow a different path, but *through the same currents*. In this case, the playback device will generate a new sequence that resembles the original in a manner similar to the classic sense of a variation on a theme.

The eddy/wood chip metaphor used in the previous paragraphs is useful and qualitatively correct, but it is obviously informal and inexact; the CHAOGRAPHER implementation, for example, uses computer simulations of differential equations, not water and wood. The mathematics is covered in detail in [3, 4].

Though the mechanism involved is very different, CHAOGRAPHER's results are reminiscent of some of Cunningham's aleatory processes: the computer program automati-

[2] Implementing this would be difficult, requiring computer vision[1] techniques.

cally divides the dance into different-length chunks and then shuffles them, producing a variation that bears striking resemblance to the original, but is also clearly different. These results are impossible to appreciate from a written description; please see the movies on the website. Our evaluations of these results are based on detailed examination by half a dozen domain experts: movement analysts (for the ballets) and martial artists (for the kenpo karate sequences). These experts affirm that CHAOGRAPHER's results fit the genre: i.e., its ballets look balletic and its kenpo karate sequences look like kenpo — and not like shokotan karate or tae kwon do[3]. We are currently in the process of doing a larger and more scientific study of these results.

As mentioned in the previous section of this paper, both CHAOGRAPHER and a rudimentary human-figure animation package are available on our website, along with instructions on how visitors to that site can use this software to:

- play and/or download existing animations from our library
- generate new animations, and upload them to the library if desired
- use CHAOGRAPHER to generate chaotic variations on any of these movement sequences, then view and/or download them

Mathematically sophisticated users can also change the defaults and invoke CHAOGRAPHER with different chaotic systems, parameter values, and/or initial conditions.

Readers who are interested in delving deeper into the field of chaos should consult any of the dozens of good nonlinear dynamics books that are currently in print. An entertaining popular overview may be found in [9], a *Scientific American*-level review appears in [2], and those with more mathematical background may find [10] interesting.

[3] These genres use many of the same postures as kenpo, but in very different sequences.

3 Learning the Grammar of Dance

One problem with any choreographic technique, automated or not, that involves subsequence reordering is that the transitions at the subsequence boundaries can be quite jarring. Figure 1, for example, shows a short section of a chaotically generated variation on a ballet adagio. Note the abrupt transition between the fifth and sixth moves of the variation. In order to "smooth" these abrupt transitions, we developed a program called MOTIONMIND, which uses *machine learning* techniques to generate stylistically consonant "tweening" sequences.

Machine learning[5] researchers are interested in constructing computer programs that automatically improve with experience, such as

- robots that learn their way through mazes
- autonomous vehicles that learn to drive on highways, or
- computer programs that ingest a few hundred issues of the Wall Street Journal and learn the rules of English grammar

The MOTIONMIND program[11] is a machine learning tool for dance. Its input is a *corpus* of movement sequences (e.g., a group of ten Balanchine ballets, animated in Life Forms or some equivalent). Using statistics

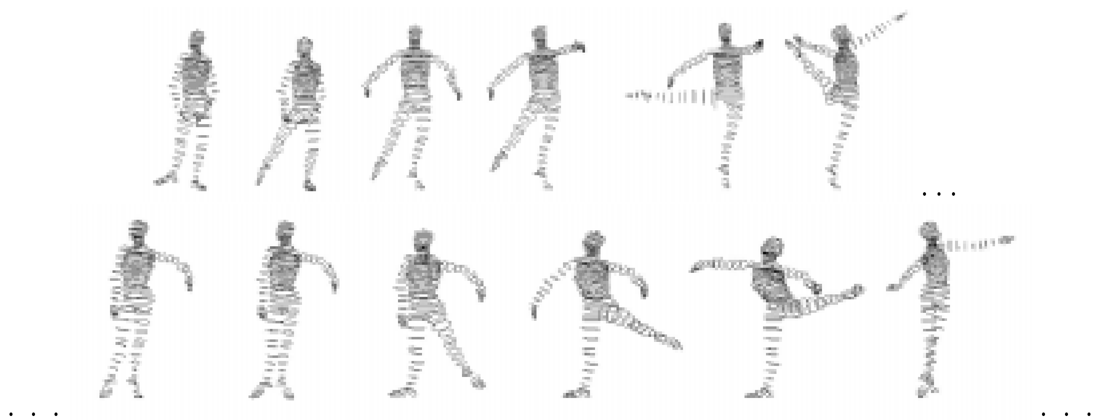

Figure 1: Part of a ballet adagio (top) and a short section of a chaotic variation on that sequence (bottom) that was generated by the CHAOGRAPHER program. Note the abrupt transition between the fifth and sixth frames of the variation. MOTIONMIND can be used to smooth such transitions in a kinesiologically and stylistically consistent fashion.

and graph theory, MOTIONMIND learns the "grammar" of that movement genre — that is, what kinds of body postures tend to follow one another, and in what order.

Once the stylistic rules that are implicit in a given body of dance phrases have been captured, there are a variety of interesting ways in which one can exploit that knowledge. Among other things, one can prescribe a starting posture and an ending posture and use MOTIONMIND's knowledge base to construct a "natural" — and original — sequence that a practitioner of that genre would follow to move between those two positions.

MOTIONMIND's learning process involves examining the corpus, one joint at a time, and storing the typical patterns in which that joint moves. Our internal representation of the human body, like Life Forms's, is based on quaternions; we use four numbers to describe the position of each of the 44 main joints. To capture the movement patterns of the right knee, for instance, MOTIONMIND looks through all the movement sequences in the corpus and extracts the `right-knee` position from each posture in each sequence. It then notes what `right-knee` positions precede and follow each `right-knee` position; if the knee is bent at a 30 degree angle, for example, the preceding position is likely to be either 29 degrees or 31 degrees. The stylistic genre of the corpus is reflected by the *probabilities* of transitions between successive positions. A high jumper, for instance, will *un*bend his or her knee much faster that s/he *bends* it, so the typical progression of that joint angle will be something like $\{33 \mapsto 32 \mapsto 31 \mapsto 30 \mapsto 29 \mapsto 28 \mapsto 32 \mapsto 40 \mapsto 50 \mapsto 70...\}$, and the corpus will contain many $31 \mapsto 30$ transitions and very few $30 \mapsto 31$ transitions. MOTIONMIND keeps track of these statistics, and stores all of this information in a collection of 44 *joint transition graphs*, each of which describes the typical movement patterns of one joint. Figure 2 shows a joint transition graph for the hips that was constructed in this fashion from a corpus of 38 ballet sequences totaling 976 positions. This data structure that is known in formal terms as a *weighted directed graph* or a *hidden Markov model*; please see [11] for the technical details.

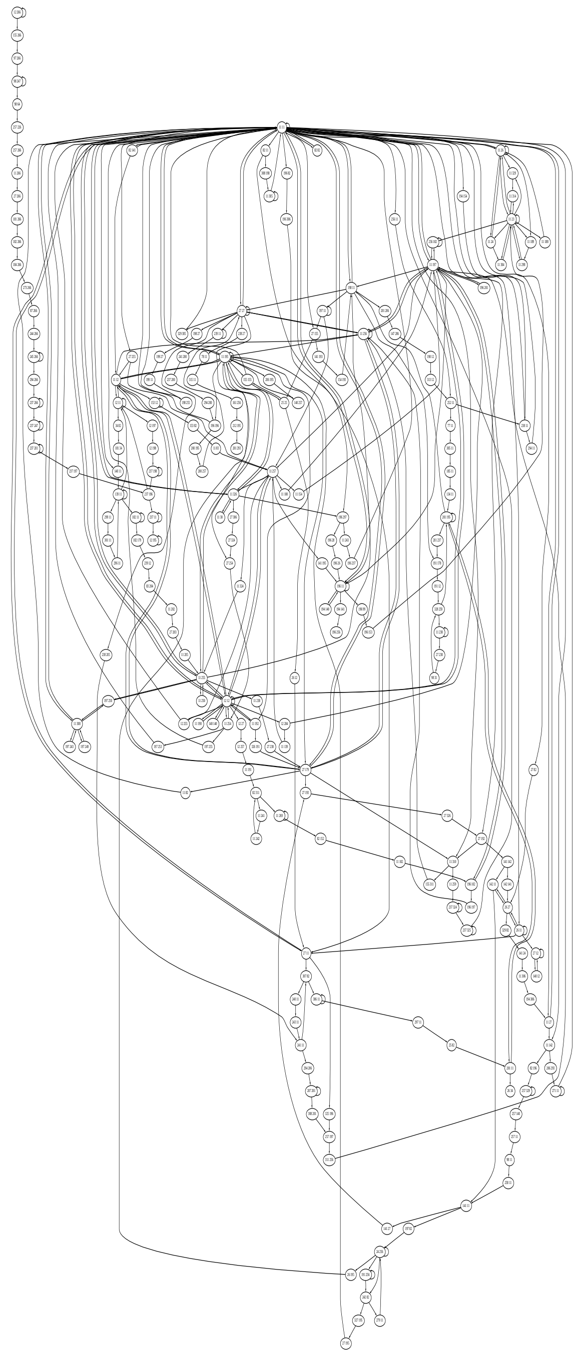

Figure 2: A mathematical scheme for the representation of movement: a *joint transition graph* that represents the movement patterns of the hips in a small corpus of 38 short ballet sequences. Each circle is a hip position, and each arrow represents a transition between the two corresponding positions.

Once MotionMind has distilled the movement patterns in a corpus into 44 joint transition graphs, it can use that information to generate original dance sequences that fit the genre of that corpus. We have explored this idea extensively, using a corpus of 38 short ballets comprising 976 distinct body postures. Once MotionMind had learned the patterns in this corpus, we presented it with a variety of "tweening" tasks, giving it arbitrary starting and ending postures and requesting that it search through the graphs that it had constructed and create a sequence joining those two postures. Because MotionMind's graphs reflect the structure of the corpus, that newly generated movement sequence retains the stylistic stamp of the original material.

Figure 3 shows an example of such a sequence. The starting and ending body postures (top left and top right in figure 3, labeled $\boxed{1}$ and $\boxed{10}$, respectively) are quite different; note the facing of the dancer and the weight distribution on the feet, for example. The eight-move sequence computed by MotionMind moves between those positions in a very natural way. The program's first move, for instance, is to lower the left leg, a natural strategy if one is going to change one's facing and end up on two feet. The following move is a simple weight shift (frames $\boxed{4}$ and $\boxed{5}$), in preparation for a lift of the right leg. This lift, which is not strictly necessary to move from the fifth frame to the tenth, is an innovation that MotionMind created based on the patterns that it observed in the corpus; it reflects the fact that ballet dancers rarely spin with *both* feet flat on the ground. Perhaps the most interesting thing about this interpolation sequence, from a balletic standpoint, is the relevé that MotionMind used during the direction shift between frames $\boxed{6}$ and $\boxed{10}$. Many relevés appear in the corpus, but none of them is associated with upper body positions that resemble the one that appears in this sequence. In using a relevé in a different (and appropriate) context, MotionMind invented a physically *and stylistically* appropriate way to move the dancer between the specified positions. The sequence in figure 3 includes a variety of other stylistically consonant innovations as well; consider, for example, the uplifted chest and chin in frames $\boxed{7}$ and $\boxed{9}$. Recall that these postures were not simply pasted in verbatim from the corpus; most of them never appear in the corpus at all. Rather, MotionMind synthesizes them *joint by joint*, using the machine learning algorithms described above, and their fit to the genre is strong evidence of the success of these methods.

Again, these results are impossible to appreciate from a static description; please see our website for a variety of movies of MotionMind's sequences.

The whole point of the machine learning procedure is to capture the patterns in a corpus, so the composition of that corpus obviously affects its results. If MotionMind is given two dozen balletic works and one short gymnastics routine, for instance, the sequences that it constructs by learning and using the grammar of that corpus will occasionally startle the observer. Corpus size is also an issue; 976 postures is an extremely meager sample of human body motion, and the knowledge that MotionMind captures from such a corpus is necessarily idiosyncratic. This causes the program to make some mathematically and/or aesthetically interesting "mistakes." If a particular elbow angle only appears once, for instance, as part of a single arm sequence that is very different from anything else in the corpus, then MotionMind is forced to use that *entire sequence* in order to make use of that elbow angle. The problem is that the graph

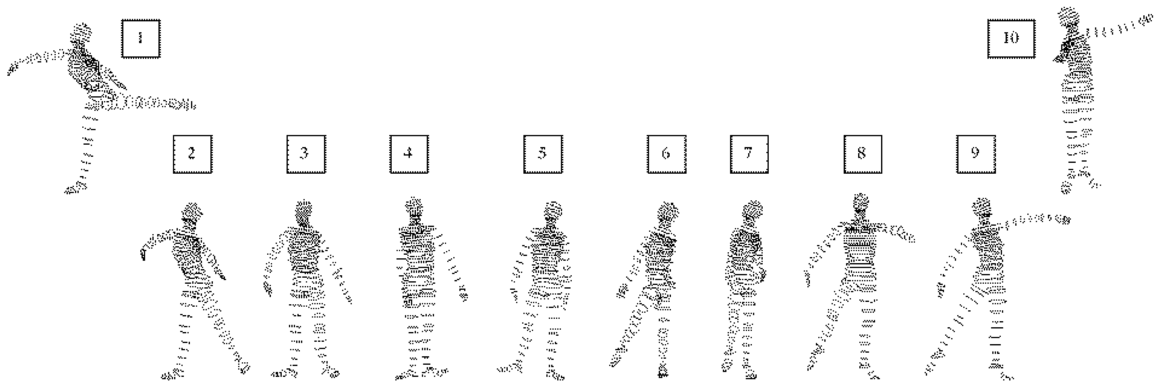

Figure 3: A "tweening" sequence computed by the MotionMind program. Inputs are the starting and ending positions shown at the top left and top right, respectively; the eight frames below them were computed automatically by the program.

is very loosely connected, the solution is to increase the richness of the corpus, and this was a large part of the motivation for our development effort on the website. We encourage all interested readers to visit the site; any and all animations that anyone is willing to contribute (by uploading them to our library) would be a real aid to this project.

4 Mathematical Movement Analysis

Dance is, in essence, motion and, therefore, transitory. While videotape has enhanced the analytical possibilities available to dancers, there is still much that goes by too quickly. It would be useful, at times, to "stop the world" and examine more subtle components of motion, including positions, angles, and speeds. A similar need arises in mathematics and physics teaching in middle and high school, where analyzing motion is a way for students to understand the relationships among position, velocity, acceleration, and to approach the fundamental concepts of calculus. *Video-based lab* research[7] seeks to support these students' understanding of the relationship between visible motion and the graphs that represent its mathematical essence. We would like to suggest that this kind of software might also be very useful in dance analysis.

The user of a video-based lab is involved in the process of analyzing motion from the very beginning; the computer is a tool, but not a dance expert, so the choice of focus is the user's, not the computer's. In order to develop a connection between motion and mathematical representation, a user starts with digitized video of some interesting motion — say, a cartwheel — and chooses a point to follow — say, the left hand. By indicating the location of the point of interest (with a mouse click) in each video frame (i.e., potentially 30 times/second), the user constructs a graph that represents the changing position of the hand. With this information, it is simple for the user to see when the hand is furthest from the body, when it is at a particular angle from vertical, or when it is directly across from the other hand. It is also possible for the software to calculate the speed at which the hand is moving and to present a graph of this information.

Much of the power of video-based labs comes from the unique way they connect the video with the graphs. The user can manipulate the video (run it forward or backward, stop at a particular frame, etc.) and the corresponding points on the graph will be highlighted. Even more interesting is a feature that allows the student to point to a particular place on the graph — perhaps the one that shows the hand moving very quickly — and see the part of the video where the corresponding action takes place. It is also possible to compare two videos where the same motion is ostensibly taking place and use the graph/video combination to pinpoint and analyze similarities and differences between the two performances.

CamMotion[8] is one such VBL, developed for high-school students, with as yet unrealized potential for the analysis of dance movement. Some of the uses we anticipate for this tool are listed below.

- In dance training:
  - as a comparative assessment tool for specific movement patterns
  - for demonstration of the role of positional exactitude and its relationship to kinetics
- In stylistic analysis:
  - for objective measurement of both position and action characteristics and proportions
- In rehearsal:
  - for clarification of movement specifics
- In creative applications:
  - for generation of film/video artifacts to accompany live movement

We are currently exploring some of these ideas and will report on the results in a future paper.

5 Conclusion

The goal of this paper is to convince dancers that they should be deeply interested in computers — and not just as recording, visualization, or special-effects aids. Computers can be active collaborators in the generation of stylistic dance phrases as well as in the mathematical description of deep characteristics of movement patterns. The potential synergy between sophisticated computer techniques and modern approaches to dance goes far beyond these uses — and most likely further than we can currently imagine. The projects that we describe here provide some idea of the forms those mutually beneficial interactions might take. In particular, we have described how research results from several cutting-edge areas of computer science — machine learning, artificial intelligence, chaos theory, and digital video analysis — can make useful and interesting contributions to the learning, inspiration, and analysis of dance.

Acknowledgements: Josh Stuart designed and implemented the CHAOGRAPHER and MOTIONMIND programs.

References

[1] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, 1982.

[2] E. Bradley. Causes and effects of chaos. *Computers and Graphics*, 19:755–778, 1995.

[3] E. Bradley and J. Stuart. Using chaos to generate choreographic variations. In *Proceedings of the Fourth Experimental Chaos Conference*, pages 451–456, 1997.

[4] E. Bradley and J. Stuart. Using chaos to generate variations on movement sequences. *Chaos*, 8:800–807, 1998.

[5] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[6] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.

[7] A. Rubin. Video laboratories: Tools for scientific education. *Communications of the ACM*, 36(5):64–65, 1993.

[8] A. Rubin. Cartwheeling through CamMotion. *Communications of the ACM*, 39(8):84–85, 1996.

[9] I. Stewart. *Does God Play Dice?: The Mathematics of Chaos*. Blackwell, Cambridge MA, 1989.

[10] S. Strogatz. *Nonlinear Dynamics and Chaos*. Addison-Wesley, Reading, MA, 1994.

[11] J. Stuart and E. Bradley. Learning the grammar of dance. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1998.

[12] P. Winston. *Artificial Intelligence*. Addison Wesley, Redwood City CA, 1992. Third Edition.

Biographies:

Elizabeth Bradley received the S.B., S.M., and Ph.D. degrees from M.I.T. in 1983, 1986, and 1992, respectively, including a one-year leave of absence to compete in the 1988 Olympic Games, and has been on the Computer Science faculty at the University of Colorado since 1993. She is the recipient of a National Young Investigator award and a Packard Fellowship.

David Capps spent fifteen years dancing, teaching, and choreographing in New York City before joining the University of Colorado Dance faculty in 1993. His choreography has appeared at the Lied Center (Lawrence, Kansas), The Changing Scene (Denver), and in New York at St. Mark's Danspace Project, the 92nd Street Y, and the Cunningham Studio.

Andee Rubin is a Visiting Scientist in the Department of Computer Science at the University of Colorado. She has worked extensively in mathematics education, developing software and curriculum in statistics for middle and high school and a complete K-5 curriculum. She is the designer of TapeMeasure and CamMotion, two ground-breaking experiments in using digital video in math and science education.