Using Chaos to Generate Variations On Movement Sequences

Elizabeth Bradley
Joshua Stuart

CU-CS-844-97

University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

Using Chaos to Generate Choreographic Variations

Elizabeth Bradley and Joshua Stuart[1]

Department of Computer Science
University of Colorado
Boulder, Colorado, USA 80309-0430
[lizb,stuartj]@cs.colorado.edu

Department of Computer Science
Technical Report CU-CS 844-97

In review for *Chaos*

Abstract

We describe a method for introducing variations into predefined dance sequences using a chaotic mapping. A symbol sequence representing a dance piece is mapped onto a chaotic trajectory, establishing a symbolic dynamics that links the dance sequence and the attractor structure. A choreographic variation on the original piece is created by generating a trajectory with slightly different initial conditions, inverting the mapping, and using special corpus-based interpolation schemes to smooth any abrupt transitions. Sensitive dependence guarantees that the variation is different from the original; the attractor structure and the symbolic dynamics guarantee that the two resemble one another in both esthetic and mathematical senses.

1 Introduction

This paper describes a chaotic mapping technique that creates variations on predefined dance sequences. A sequence of specialized symbols representing the body positions in a dance piece is mapped onto a chaotic attractor, establishing a symbolic dynamics that links the dance progression and the attractor geometry. We then use this mapping to create a choreographic variation by generating a new trajectory — with the same dynamic system and slightly different initial conditions — and then inverting the mapping. Sensitive dependence guarantees that the variation is different from the original; the attractor structure and the symbolic dynamics guarantee that the two resemble one another in both the esthetic and the mathematical senses.

To establish the mapping between an N-move dance sequence, like the one shown in figure 1, and a chaotic attractor, we first integrate a chaotic ODE system $\dot{x} =$

[1]Supported by NSF NYI #CCR-9357740, ONR #N00014-96-1-0720, and a Packard Fellowship in Science and Engineering from the David and Lucile Packard Foundation.

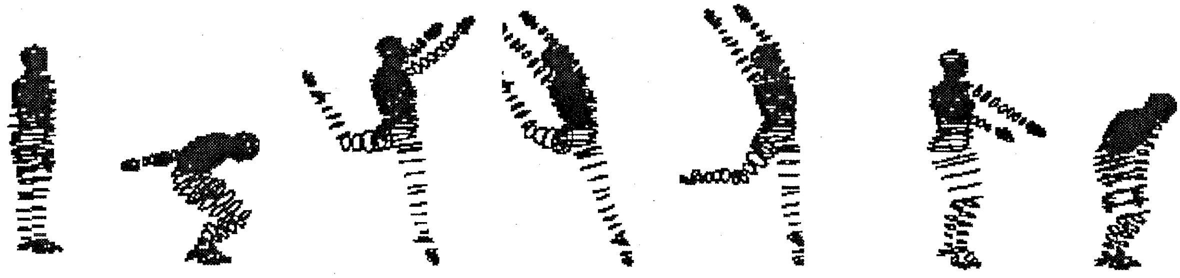


Figure 1: A seven-position ballet jump sequence

$f(x)$, $x(t) \in R^n$ numerically from some initial condition x_0. We then use a Voronoi diagram to partition the state-space region occupied by the ω-limit set $\phi_\omega(x_0)$ of this trajectory into N cells. Finally, we label the *itinerary* of cells traced out by $\phi_\omega(x_0)$ with special symbols that represent the sequence of body positions in a predefined dance piece. To create a variation, we then generate a new trajectory $\phi(x')$ from an initial condition x' near the attractor and invert the mapping: at each timestep, the dance move corresponding to the cell in which $\phi(x')$ falls is sent to an animation tool.

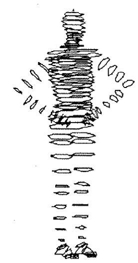
This work was catalyzed by a similar scheme, proposed by Diana Dabby[2, 3], that uses a related mapping to generate *musical* variations. The core idea here is the same, but many of the issues and tactics — together with much of the mathematics — are very different. The symbol set is one obvious distinction. There is a simple, well-established notational scheme for music, but body positions are much harder to represent; we use representational techniques from rigid-body mechanics to solve this problem. The mathematics of the mapping is also very different; Dabby uses a simple metric on a projection of a transient trajectory to define cells, whereas we work with a full, formal symbolic dynamics *on the attractor*, derived using computational geometry techniques. Finally, while musical instruments can play arbitrary pitch sequences, kinesiology and dance style impose a variety of constraints on consecutive body postures. To address this problem and smooth any abrupt transitions introduced by the chaotic mapping, we have developed a class of corpus-based interpolation schemes that capture and enforce the dynamics of a given dance genre.

The results produced by these mapping and interpolation algorithms have intrigued both dynamicists and dancers. The chaotic variations bear an obvious resemblance to the originals[2], and yet they are also clearly different; broadly speaking, the variations resemble the originals with some shuffling of coherent subsequences. When contrasted with random shuffles of the same sequences, the properties of this scheme become even more apparent: the randomized "variations" bear no temporal resemblance to the original whatsoever. It is impossible to appreciate these results from a textual description; please see the animations at `http://www.cs.colorado.edu/~lizb/chaotic-dance.html`.

[2] A well-known dynamicist opined "It looks like Al Gore doing the macarena."

```
{pelvis,       0.37187, 0, 0, 0.92823;
 atlantal,     -0.16417, 0, 0.00012, 0.98626;
 cervical_1,   0, 0, 0, 1;
 ...
 left_toes,    0, 0, 0, 1 }
```

(a)



(b)

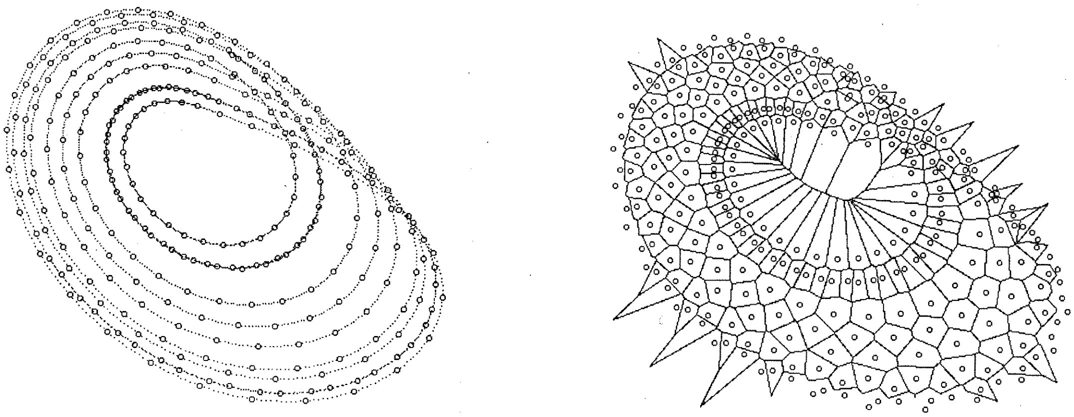
Figure 2: Symbolic representation of the human body: (a) the descriptor/quaternion symbol, which specifies a vector and an angle of rotation around it for each of the 44 main joints in the body (b) the corresponding graphical representation used by the Life Forms animation tool.

2 Linking Attractor Geometry and Dance Structure

2.1 Symbolic Dynamics and Body Positions

A point in a state-space trajectory of a dynamic system can be described at different precisions, ranging from a tuple of real numbers to a symbol that identifies a large state-space region. Though the coarse-grained nature of the latter abstracts away much detailed information about the dynamics, it preserves many of its invariant properties; see, e.g., Hao[7] for details. Establishing such a *symbolic dynamics*[10] presents two problems: the partition and the ordering. This paper offers novel and unusual solutions to both: we use computational geometry techniques on points of a trajectory to obtain a good partition, and we use the natural progression of body positions in a dance sequence to induce the symbol order.

The symbol set used in our algorithms represents the position of each of the 44 primary joints in the human body with a *quaternion* — a standard representation in rigid-body dynamics, dating back to Hamilton[6]. A quaternion $Q(r, \vec{u})$ consists of a three-space vector $\vec{u}$ and a scalar r that specifies the angle of rotation around that vector. Thus, a body position symbol is quite complicated: 44 descriptors (`pelvis`, `right-wrist`, etc.), 176 floating-point numbers (four for each joint), and a variety of information about the position and orientation of the center of mass. See figure 2 for an example. This complexity is simply a reflection of the representational task involved; Labanotation[8], the graphically intricate system used by professional dance notators, is even more baroque: attaining proficiency in its use requires years of practice.



(a) (b)

Figure 3: A Rössler trajectory for a 300-move dance sequence and the associated Voronoi-diagram partition of the attractor. The perpendicular bisectors of the circled trajectory points in (a) yield the Voronoi diagram in (b). Line segments that extend beyond the bounding box of part (b) have been omitted from this plot.

2.2 Tiling the Attractor

Creating a partition for the purposes of re-mapping a dance sequence requires tiling the state-space region occupied by the attractor with N nonoverlapping cells, where N is the number of moves in the sequence. To accomplish this, we first integrate a chaotic ODE system $\dot{x} = f(x)$, $x(t) \in R^n$ with 4^{th}-order Runge-Kutta[12] from some initial condition x_0 and let the transient die out. The requirements of the mapping limit the number of cells to N, but the partition requires that the collection of cells cover the attractor, and spurious numerical effects preclude simply increasing the time step until a fixed-length (N-point) trajectory covers a given attractor. We address this by fixing the timestep and trajectory length[3] and using a "skip" parameter, m, to control the spacing of the trajectory points that are actually used to construct the cells. In figure 3, for instance, $m = 10$, so every tenth trajectory point (shown circled) generates a cell. The specific algorithm that we use to actually construct the cells, the Voronoi diagram[11], is drawn from computational geometry; in it, one constructs the perpendicular bisector of every adjacent pair of points and intersects them, as shown in figure 3. Note that a Voronoi diagram is essentially the dual of a Delaunay triangulation. The actual implementation uses K-D trees[5], rather than the usual Voronoi diagram construction algorithm, to reduce the computational complexity of the nearest-neighbor step in the algorithm from $O(N^2)$ to $O(N \log N)$.

[3]We choose values for these parameters that assure that the transient has died out, the attractor is covered, and the dynamics include no spurious numerical effects.

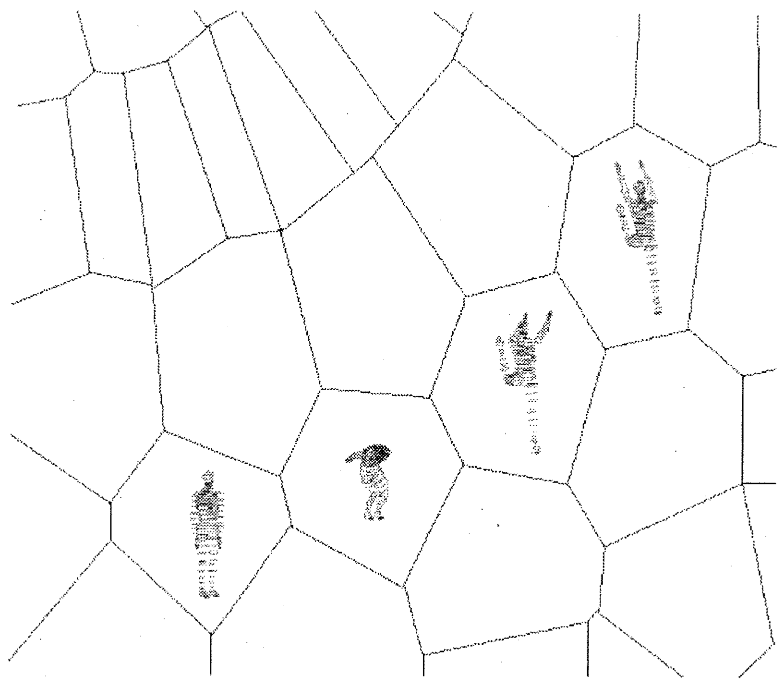


Figure 4: Part of the chaotic mapping that links a longer version of the jump sequence of figure 1 and the Rössler attractor geometry of figure 3.

2.3 Establishing and Using the Mapping

Given an N-move dance sequence, expressed in terms of the quaternion-based symbol set described in section 2.1, and an N-cell Voronoi tiling of a chaotic attractor — like the $N = 300$ example shown in figure 3 — establishing a mapping that links the attractor geometry and the dance structure simply amounts to equating indices: the first entry in the *itinerary* of cells traversed by the trajectory $\phi_\omega(x_0)$ is labeled with the symbol that describes the first position in the dance sequence, and so on[4], as depicted schematically in figure 4. Using this mapping to create a variation is equally simple, but slightly more computationally expensive: we generate another trajectory $\phi(x')$ from an initial condition x' near the attractor, use the K-D tree to determine the Voronoi cell in which its first point falls, output the associated body-position symbol to an animation tool, skip m points, and repeat to the desired variation length. For a 1000-position dance sequence, the entire re-mapping procedure requires 18 milliseconds on a PowerMac running MacOS 7.5.5; without the K-D tree, it takes 30.2 milliseconds. The K-D tree advantage grows with the sequence length: for a 9000-move sequence, the times are 156.2 and 2324.2, respectively.

[4]This representation — a symbolic dynamics induced by that dance sequence — is conjugate to the R^n dynamics of $\phi_\omega(x_0)$, which is interesting in that it implies that the animations are formally equivalent, in a precise mathematical sense, to the "real" dynamics[7, 9]

3 Interpolation

The re-mapping scheme described in the previous section introduces abrupt transitions in the chaotic variation — places where consecutive body positions would require physically impossible or stylistically illegal moves. The interpolation scheme described in this section inserts new body postures into these gaps in order to smooth the progression. A simple and obvious way to do this would be to use splines or some other purely mathematical technique on the quaternion data to manufacture new body positions to span the transition. This does not, however, address the problem of stylistic or kinesiological illegality; a spline-based interpolation may not, for instance, adhere to the requirement that ballet motion is linear or that the elbow only bends 180 degrees. To solve these problems, we use a corpus of human movement (e.g., ten Balanchine ballets) to select a sequence of postures that would naturally occur between the two positions that frame the abrupt transition.

To this end, we use the corpus to build a labeled, directed graph $G(V, E)$ that captures the dance sequence. Each body position in the corpus is represented by one vertex v_i and each transition between successive postures is represented by an edge e_{ij} between the corresponding vertices. Figure 5 shows an example of such a graph. In this formulation, an illegal transition — defined as one that is not present in the corpus — is a pair of vertices v_l and v_n that are not linked by a single edge e_{ln}, such as the postures labeled **a** and **c** in figure 5. When such a transition is encountered in the chaotic variation, we use $G(V, E)$ to compute an interpolation subsequence that starts with v_l, ends with v_n, and is consistent with the corpus. Specifically, we use a forward-backward modified Dijkstra's algorithm to find the shortest path in G between v_l and v_n, and then insert the body positions corresponding to the vertices traversed by that path into the gap in the original sequence. In figure 5, for example, there are two two-edge paths that link postures **a** and **c**: $\{\mathbf{a} \rightarrow \mathbf{b} \rightarrow \mathbf{c}\}$ and $\{\mathbf{a} \rightarrow \mathbf{e} \rightarrow \mathbf{c}\}$. The abrupt transition $\{\mathbf{a} \rightarrow \mathbf{c}\}$ could be patched with either of these two subsequences. Dijkstra's algorithm[4] finds the shortest path from a single source vertex to all other vertices in a graph; if more than one "shortest" path exists, as in figure 5, it returns the first one it encounters. For the forward version, v_l is used as the source vertex. For the backward version, v_n is used as the source and the orientations of the edges in G are reversed. The forward and backward algorithms are invoked simultaneously; each one progressively deepens its search until a common vertex v_m is encountered. The paths from v_l to v_m and from v_m to v_n are then merged to give the desired shortest path from v_l to v_n. The worst case total running time of this algorithm is $O((V + E)\log_2 V)$[1].

To better model and enforce the nuances of a particular movement style, we are improving this scheme in two ways. The first involves finer-grained physical representation and interpolation. Currently, the atomic representational unit is a full body position; the next version will perform *joint-wise* interpolation instead — e.g., bridging a gap by moving the arm from its quaternion position in v_l to its quaternion position in v_n according to the rules for *arm* movement implicit in the corpus, and so on, rather than searching for and patching in *full body positions*. The second improvement involves probabilistic

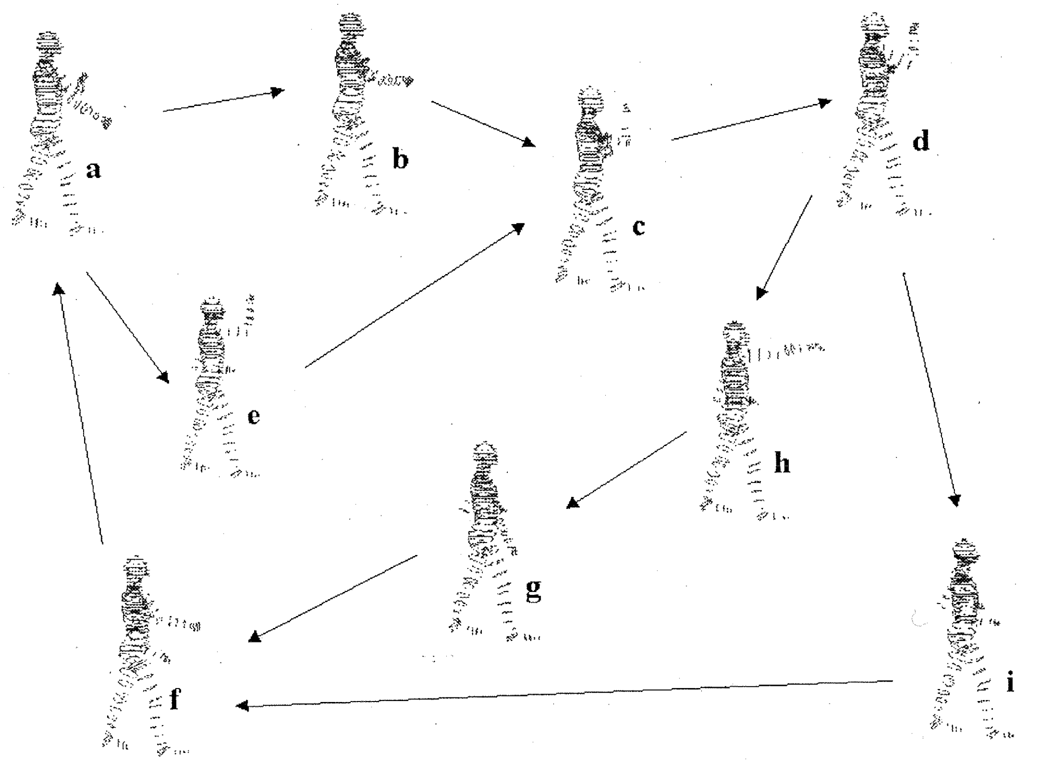


Figure 5: A labeled, directed graph representing a small corpus of human movement. Vertices represent postures observed in the corpus and edges depict movement sequences between those postures.

analysis of the transitions in the corpus. $G(V, E)$ is currently a simple labeled, directed graph, where transition legality is represented by the presence or absence of an edge; to this, we are adding edge weights that represent a measure of the probability of each inter-move transition. One logical choice for these weights is the negative log-likelihood:

$$w_{ij} = -\log(f_i) + \log(f_j) - \log(f_{ij}),$$

where f_i and f_j are the frequencies of postures i and j and f_{ij} is the frequency with which posture j follows posture i. Small values for w_{ij} correspond to transitions that are more likely to occur. With this addition, the interpolation scheme finds more-natural subsequences with which to smooth abrupt transitions. For instance, a five-edge path may have a much higher probability than a two-edge path if the latter is only observed rarely in the corpus, and adding edge weights to G allows the interpolation scheme to enforce that constraint.

Fine-grained, jointwise interpolation with log-likelihood edge weights, as described in the previous paragraph, is theoretically a good solution for the problem at hand, but its computational complexity is prohibitive. For example, if each joint can be in one of only ten possible orientations, then G could contain $O(10^{88})$ vertices. One way to manage this complexity is to use a hierarchical data structure that exploits the structure and physics of the human body — the notion, for example, that the position of the wrist strongly affects the position of the fingers but has little effect on the toes. The physical structure of the human body is depicted graphically in part (a) of figure 6. We use a tree to represent this structure in the form of dependency relationships between joints, as shown in part (b) of the figure. The pelvis is the root of this tree; three branches lead from this root to nodes corresponding to the right thigh/hip joint, the left thigh/hip joint, and a joint representing the lower spine[5]. Each hip joint is the parent node to a knee (n), and so on. Associated with each node of this tree is a graph that contains a vertex for each observed state of the corresponding joint, together with a set of edges that define how that joint reacts to movements of its parent joint. Figure 6(c) shows an example: the graph associated with the lumbar-spine node l in a tree built from a corpus where that joint takes on two orientations l_1, l_2 and the pelvis takes on three: p_1, p_2, p_3. If the lumbar spine is in position l_1 and the pelvis moves from p_1 to p_3, then the lumbar spine will move to position l_2 with the probability associated with the $p_1 \rightarrow p_3$ edge from the node l_1 in the graph in part (c)[6]. One can view this hierarchical graph structure as a set of first-order Markov chains, in which a single chain represents the orientation of each joint in the body. Each Markov chain contains a different set of state transitions and transition probabilities for each transition pair in the joint's parent node.

Many of the techniques in this section, as well as others on which we are currently working, were inspired by solutions to similar problems that arise in molecular biology (DNA sequencing) and computational linguistics (learning a grammar from a corpus and then using it to construct meaningful sentences).

---

[5]The sacrum and the five lumbar vertebrae are lumped together; this representation sacrifices some back suppleness for lowered complexity.

[6]These probabilities have been omitted from the figure for clarity.

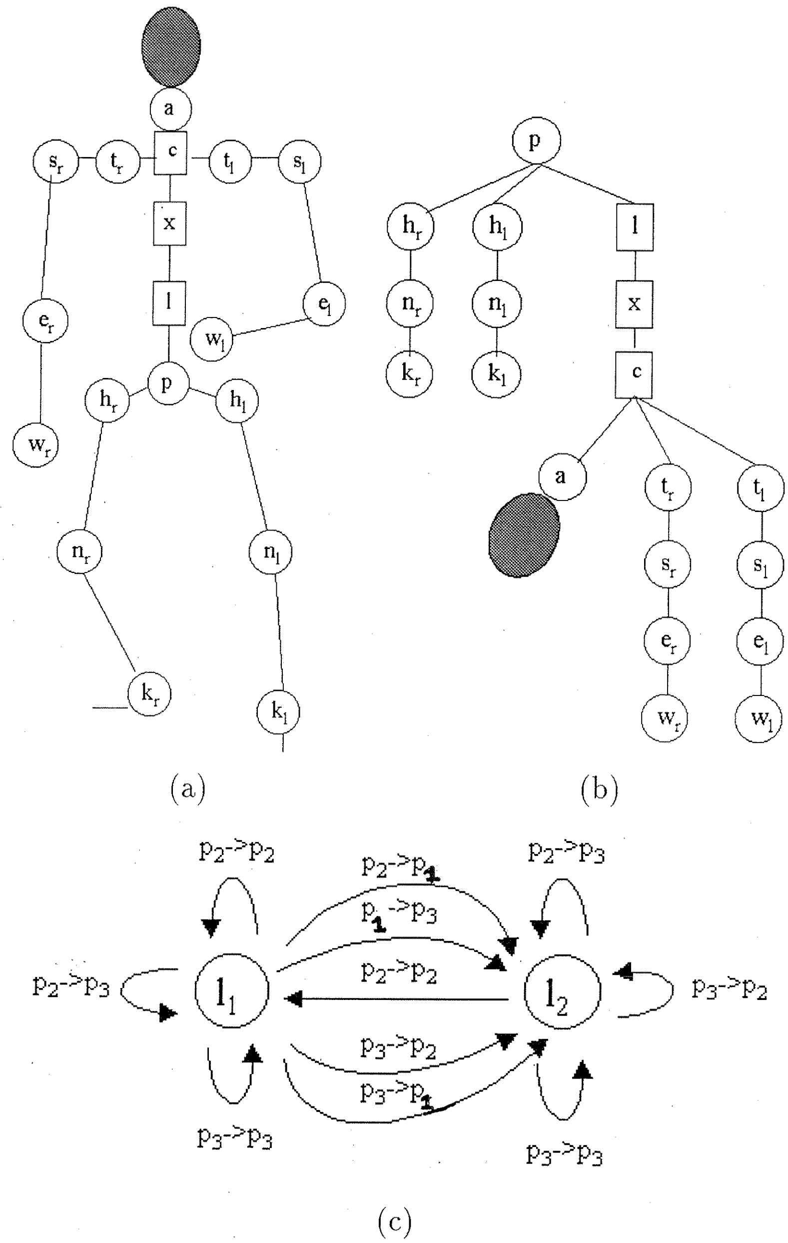


Figure 6: A hierarchical representation of the human body. The shaded ellipse is the head; toes and fingers have been omitted for clarity. Part (a) depicts the body, with all joints labeled. Bilateral joints are identified with subscripts according to the side of the body on which they fall (e.g., right and left elbows e_r and e_l). Part (b) shows the dependencies induced by the connectivity of the body: for instance, the position of the pelvis influences the positions of both hips h_r and h_l and the lumbar spine l. The graph in part (c), associated with the lumbar spine node l, captures how that joint reacts to movements of its parent joint p, the pelvis.

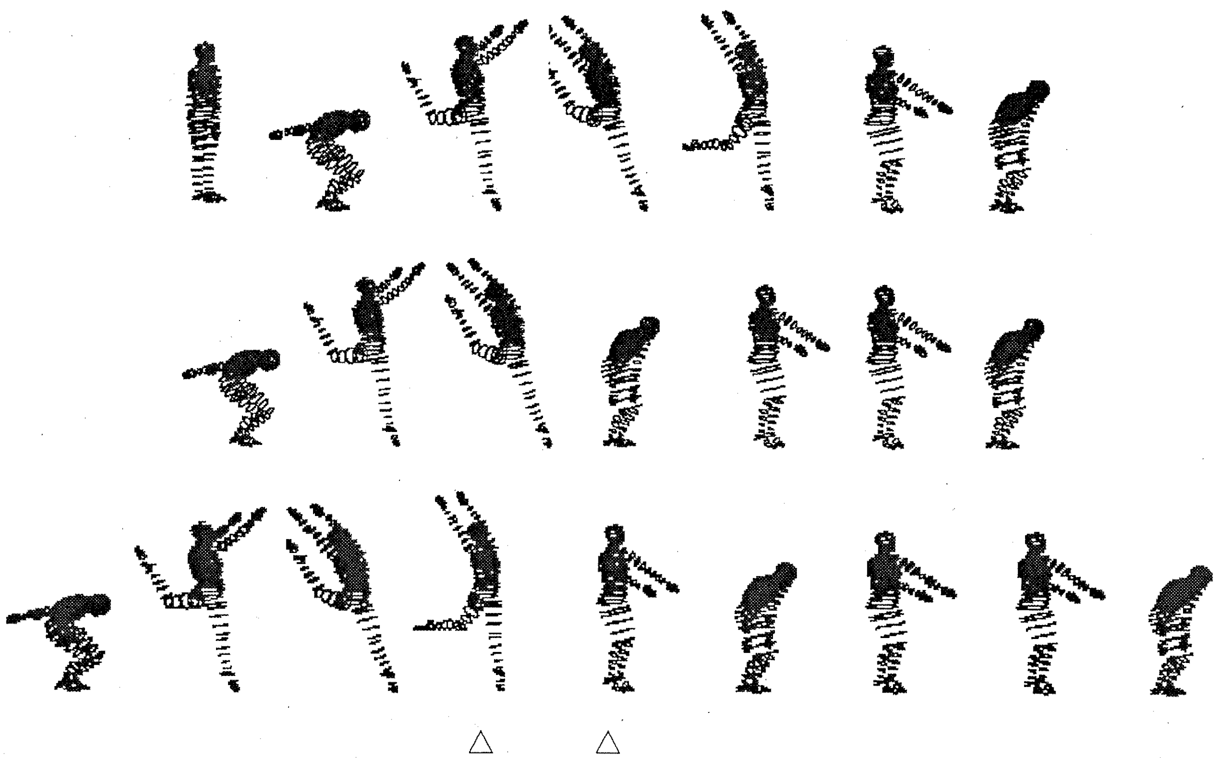


Figure 7: A ballet jump: the original sequence (above), a chaotic variation on that original (middle), and an interpolated version of that variation (below). The moves identified by arrows in the lower sequence were inserted by the interpolation scheme to smooth an abrupt transition between the third and fourth moves in the chaotic variation above it.

4 Results

Figure 7 shows the simple ballet jump sequence of figure 1, a chaotic variation of that jump generated with the Lorenz system, and a smoothed version of that variation. The sequence shown in the middle row of figure 7 was derived from the original using the re-mapping scheme described in section 2. An abrupt transition is visible between the third and fourth moves of this variation; the corpus-based graph-theoretic interpolation scheme described in section 3 inserted two new moves to produce the smoothed sequence shown at the bottom of the figure. Note that the inserted moves define a very natural way to move between the two body positions that frame the abrupt transition.

While it is clear from the figure that the jump positions are indeed shuffled and that the interpolated version is indeed smoother, it is impossible to appreciate these results from a static portrayal of such a short sequence; please see the web site listed at the end of the introduction for a variety of animated variations — the jump shown above, a popular dance progression (the macarena), a martial arts "form" drawn from the discipline of kenpo karate, and a medley of all three of these movement sequences. Variations were constructed on each piece using two different chaotic systems (Lorenz and Rössler) in order to show how the attractor geometry affects the variation. Loosely speaking, the variations resemble the originals with some shuffling of coherent subsequences; this is most obvious in the medley, where the variation clearly shifts back and forth between genres. Where there is an obvious genre, such as the karate sequence, the variation fits that genre. In fact, the point of using this sequence was the distinct, well-defined structure of individual martial arts genres, and the goal was to determine whether variations generated on kenpo karate sequences still looked, to the expert eye, like kenpo — and not like shokotan karate or tae kwon do. We also present *randomly* shuffled versions of each of the four pieces in order to demonstrate, by contrast, how much structure is retained by the chaotic variation scheme. Perhaps the most telling comparison is between the chaotic and randomized versions of the medley; segments of the individual dances are clearly visible in the former and utterly absent in the latter.

These results set off a variety of interesting questions. For instance, a shorter dance sequence implies larger cells and hence a "coarser" symbolic dynamics; this has interesting effects on how smoothly the cell itinerary of $\phi(x')$ moves and shifts along the original attractor, with corresponding implications for the animation and its resemblance to the original piece. The attractor geometry plays a mathematically and visually obvious role in the character of the variation; note the differences between the Lorenz and Rössler pieces on the web site. It appears that the latter contains longer coherent original subsequences than the former, which is consistent with the Möbius-band nature of the Rössler attractor, in comparison to the bilaterally symmetric two-lobed Lorenz geometry. We are in the process of performing a statistical analysis on the two pieces in order to determine whether these patterns are real or illusory.

Besides the animations and the associated explanation and analysis, the web site also contains a simple animation package and the re-mapping code itself. We encourage the readers (and their students) to create new animations and/or try different ODE systems, initial conditions, and so on. New animations are particularly useful and extremely welcome; the dance world has not yet embraced the notion of computer animation, so the current critical limitation in this project is the inadequacy of the existing corpus — on which our interpolation scheme depends[7].

5 Conclusion

Evaluation of these results is necessarily subjective. We have shown these animations to hundreds of people, including dozens of dancers. The consensus is that the variations not only resemble the original pieces, but also are in some sense pleasing to the eye. They are both different from the originals and faithful to the dynamics of the dance genre: there are no jarring transitions or out-of-character moves. This is a non-trivial accomplishment. A previous attempt to use mathematics to generate choreographic

[7]The remapping code does not yet handle center-of-mass interpolation smoothly, so movement sequences where the body moves from place to place will appear choppy. We are currently working on this, but the issues involved — kinesiology, in particular — make it quite difficult.

variations — a subsequence randomization scheme used by the well-known choreographer Merce Cunningham — met with an extremely negative reception in the dance world.

From a scientific viewpoint, this scheme is interesting for several reasons. It involves a formal (albeit unusual) application of symbolic dynamics, the properties of chaotic attractors, and rigid-body mechanics: the partition for the symbolic dynamics is generated automatically using computational geometry techniques and the natural order of the dance sequence, and the symbol set relies on a representational device invented by Hamilton himself. By applying methods from graph theory, statistics, and computational linguistics to a corpus of dances from a particular genre, the interpolation scheme proposed here smooths awkward transitions in a physically *and stylistically* coherent fashion. Last, but certainly not least, showing these results in a classroom is an enormously effective way to motivate students to learn the mathematics of rigid-body dynamics and chaos.

Acknowledgements: Apollo Hogan contributed the K-D tree code and a variety of good ideas to this project and Meenakshy Chakravorty helped set up the website and write the associated animation code. The authors would like to thank both David Capps of the University of Colorado Department of Theater and Dance and Dan Jurafsky of the University of Colorado Department of Linguistics for helpful conversations, and Margo Seltzer of Harvard University for suggesting and demonstrating the karate sequences.

References

[1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* The MIT Press, 1990. pp 527-531.

[2] D. S. Dabby. Musical variations from a chaotic mapping. *Chaos*, 6:95–107, 1996.

[3] D. S. Dabby. A chaotic mapping for musical and image variation. In *Proceedings of the Fourth Experimental Chaos Conference*, 1997.

[4] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[5] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3:209–226, 1977.

[6] H. Goldstein. *Classical Mechanics.* Addison Wesley, Reading MA, 1980.

[7] B.-L. Hao. Symbolic dynamics and characterization of complexity. *Physica D*, 51:161–176, 1991.

[8] A. Hutchinson. *Labanotation.* The Dance Notation Bureau, NY, NY, 1970.

[9] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, 1995.

[10] M. Morse and G. A. Hedlund. Symbolic dynamics. *Am. J. Math.*, 60:815, 1938.

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

[12] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, Cambridge U.K., 1988.