Optimization and Human Movement

Elizabeth Bradley
Department of Computer Science
University of Colorado
Boulder CO 80309-0430 USA
lizb@cs.colorado.edu

Joshua M. Stuart
Stanford Medical Informatics
Stanford University Medical Center
Stanford, CA 94305-5479 USA

Human movement is an ongoing optimization process. A baseball player attempts to contact a ball with a bat so as to propel the latter as far as possible; a rower tries to impart the maximum possible force to the water through an oar, while at the same time avoiding any disturbance to the boat's forward motion. The cost functions involved are complex, implicit, nonunique, fuzzy, and often subjective. Hundreds of muscles—most of which are not under conscious control—are involved in every motion, and different body geometries, coaches, choreographers, etc., prescribe different criteria for optimal movement. At the same time, the results are unmistakeable: the difference between breakdancing and ballet is patently obvious, even to the untrained eye.

Automatic generation of motion sequences is an interesting problem that has applications in graphics and animation, in training sequence generation, in gait analysis, and even in artistic innovation[2, 3, 7]. The moving picture industry, needless to say, has devoted tremendous numbers of cycles to this problem, but the algorithms involved rely on human animators to specify "keyframes" that act as skeletons for the movement. One can use mathematical interpolation techniques like splines to move individual body parts from one keyframe to another, but these kinds of methods do not address the problem of kinesiological illegality (e.g., that the knee only bends 180 degrees, or that arms cannot pass through ribcages). Many animation packages, such as Life Forms or Poser[1], use an augmented spline approach that relies on a table of kinematic constraints to avoid illegal movements, but this type of approach is somewhat *ad hoc*. One can also generate movement sequences by modeling the physics of the body—e.g., using differential equations and solving the corresponding boundary-value problem[9]. Physics-based animation approaches are extremely interesting and highly promising, but also very difficult; deducing the control equations that humans use to recover their balance after a jump, for example, is a Ph.D. thesis-level problem[13]. *Stylistically* faithful interpolations are even harder to implement; neither splines nor $F = ma$ can easily capture or enforce, for instance, the requirement that classical ballet emphasizes position over motion[2], and developing a mathematics- or physics-based approach that does so would be all but impossible.

In this short paper, we describe an alternative solution to the "tweening" problem: a class of corpus-based schemes that generate physically consistent and stylistically consonant movement sequences between pairs of specified body positions. The computer program MotionMind, which instantiates these ideas, takes as input a corpus of movement sequences—e.g., ten Balanchine ballets—and a pair of body postures A and B; its output is a movement sequence that starts at A, ends at B, and fits the style of the corpus. If A and B are are "far apart," as measured by some metric that takes into account both the physics of the human body and the style of the movement genre, this can be nontrivial. MotionMind solves this problem by using statistical and graph-theoretic techniques to "learn" the grammar that is implicit in the corpus, and then applying simple heuristic search methods to the resulting graphs in order to generate movement sequences that are consistent with that grammar.

MotionMind simplifies the complex task of representing human motion by disregarding limb length. Each body posture is represented by a set of 23 quaternions—a common representational device in graphics that consists of a 3-vector and an angle of

[1] fas.sfu.ca/lifeforms.html and www.metacreations.com/products/poser3/

[2] In ballet, body parts tend to describe piecewise-linear paths through space, emphasizing the positions at the junctions of those linear segments; in modern dance, on the other hand, the motion *between* the endpoints is often the important feature, and the choreography is crafted accordingly.

rotation around that vector[8]—each of which specifies the position of one of the body's main joints (omitting, e.g., finger and toe knuckles).

To capture the movement patterns in a corpus, MotionMind examines that corpus joint by joint, building a directed, weighted graph for each one. Each vertex in these *joint transition graphs* represents a joint position (e.g., elbow bent to 10 degrees); edges represent observed transitions between the corresponding positions, weighted using the negative log-likelihood: small values correspond to transitions that are more likely to occur. An example of such a graph is shown in figure 1. The intricate patterns of human movement are reflected by the complex topology of the graph. Note that joint angle is a continuous variable, which would imply a potentially infinite number of vertices; to avoid this problem, MotionMind discretizes the quaternion space (cf., snapping objects to a grid in graphics).

After building the set of 23 *joint transition graphs* that capture the movement grammar, MotionMind applies memory-bounded A* search[11] in order to find interpolation sequences. In general, A* finds a path from an initial state to a goal state by progressively generating successors of the current state in the search, computing a heuristic score that combines the existing path length and an estimate of the distance to the goal, and then expanding on a best-score-first basis. See [12] for more details. In this problem, the initial and goal "states" are actually 23 states in separate graphs, and MotionMind needs to search all 23 graphs in parallel (for a path from the knee angle in posture A to the knee angle in posture B, another path from the ankle angle in posture A to the ankle angle in posture B, and so on). One obvious choice of scoring function—which is based upon the assumptions that the sequence should be as short as possible and that common movements should be chosen over rare ones—is to minimize the sum of the weights of the edges in the path.

The basic idea here is fairly simple, but further consideration reveals a variety of important additional constraints. One really wants the lengths of all of those paths to be roughly equal, for example, in order that the different body parts arrive at the target posture at about the same time. Moreover, the

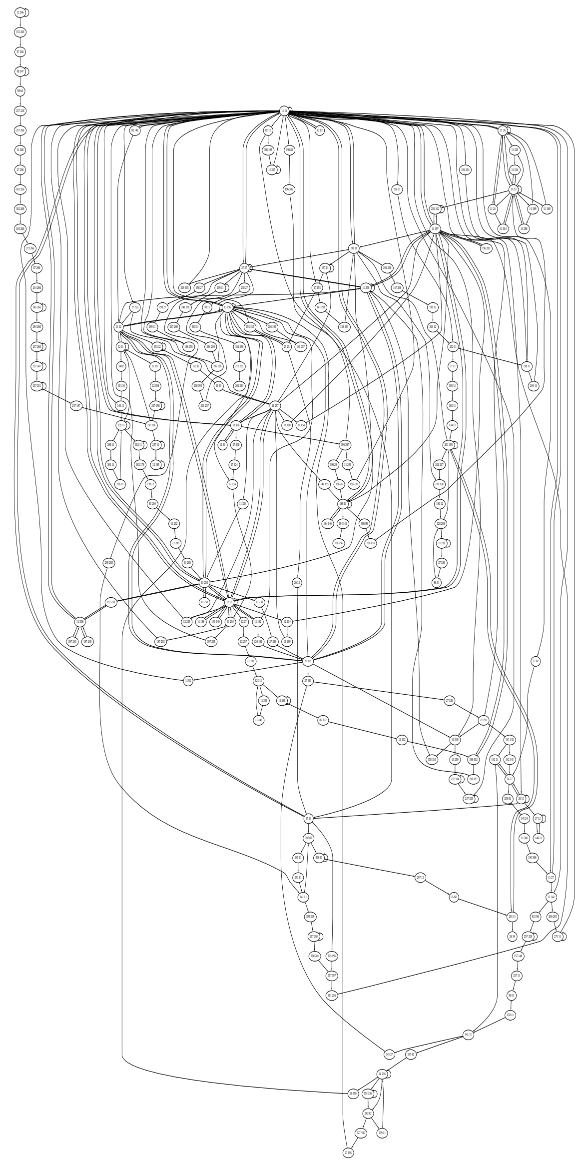
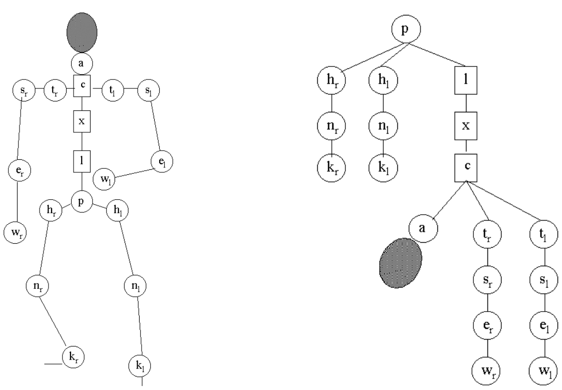


Figure 1: A *joint transition graph* that represents the movement patterns of the hips in a corpus of 38 short ballet pieces, comprising 1720 individual postures. The numbers in each state identify the discretized position of the joint. Edge weights and isolated vertices have been omitted in the interests of clarity. After [11].

search is complicated by the fact that joint positions cannot be interpolated in isolation: the movement patterns of the ankle, for instance, are strongly influenced by whether or not the foot is on the ground—information that is implicit in the positions of the pelvis, knees, etc. This requires that the expansion of nodes in the search be context dependent in a somewhat unusual way. MotionMind uses a Bayesian network[10], shown in figure 2, to model the constraints induced on joint motion by gravity and body topology. The pelvis is the root of this tree; three



(a) (b)

Figure 2: An influence diagram that explicitly represents the coordination of joints of the human body. Part (a) depicts the body and part (b) shows the inter-joint dependencies induced by gravity and topology: for instance, the position of the pelvis influences the positions of both hips h_r and h_l and the lumbar spine l, but the right and left ankles k_r and k_l do not directly influence one another. Without this simplifying assumption, the search space for this problem is intractable. After [11].

branches lead from this root to nodes corresponding to the right hip, the left hip, and the lower spine[3]. Each hip joint is the parent node to a knee, and so on. MotionMind assigns a conditional probability distribution, estimated from the corpus, to every (parent,child) pair in the tree, and models coordination by incorporating this number into the A* scoring function.

Figure 3 shows an example MotionMind sequence, computed using a ballet corpus. 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. MotionMind's eight-move interpolation sequence moves between those positions in a very natural way. Its 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 the program inserted because of the observed patterns 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 relévé[4] that the interpolation procedure inserted between frames $\boxed{6}$ and $\boxed{10}$. Many relévés appear in the corpus, but none of them are associated with upper body positions that resemble the one that appears in this sequence. MotionMind has invented a physically *and stylistically* appropriate way to move the dancer between the specified positions. The interpolation sequence in figure 3 includes a variety of other stylistically consistent innovations as well; consider, for example, the uplifted chest and chin in frames $\boxed{7}$ and $\boxed{9}$—posture elements that are quintessential ballet style. Recall that these postures were not simply pasted in verbatim from the corpus; they were synthesized *joint by joint* using the transition graphs and influence-diagram directed A* search, and their fit to the genre is strong evidence of the success of the methods described in the previous section. `mpeg` movies of this sequence, along with many others, are available on the web[1].

MotionMind's algorithms have several interesting failure modes. Because of the directed nature of the

[3]The sacrum and the five lumbar vertebrae are lumped together. This compromise sacrifices back suppleness for lowered complexity.

[4]A relévé, which consists of lifting up on one's toes, is a stylistically required component of a direction shift in ballet.

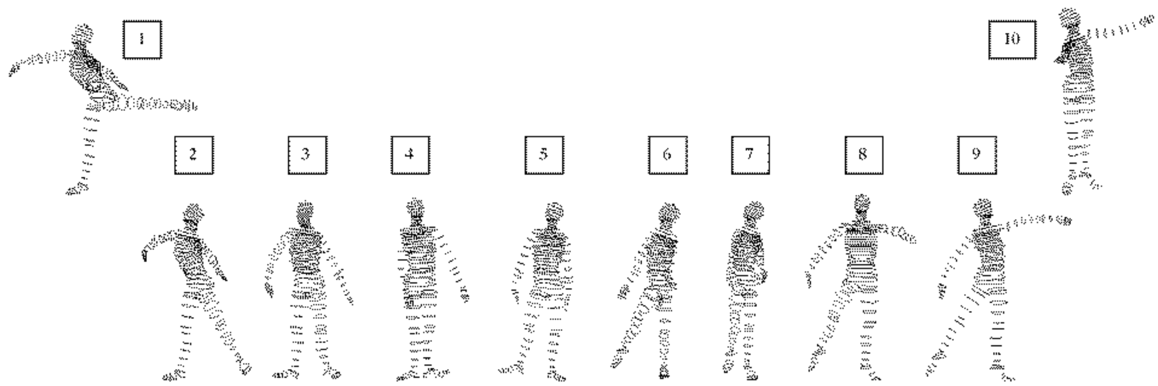


Figure 3: A "tweening" sequence generated by MotionMind. The starting and ending positions are shown at the top left and top right, respectively; the eight frames below them were computed by MotionMind. After [11].

graphs, the search algorithm sometimes has trouble finding interpolation subsequences between body positions that occur in inverted temporal order (e.g., reversing a baseball swing). Moreover, it often finds relatively long paths between positions that appear very similar; in one such instance, where the task was a simple 90-degree rotation of the right shoulder around the long axis of the arm, MotionMind constructed an 65-move sequence that involved much leg and trunk movement. Both of these problems are caused by limited corpus size. 1720 postures is an extremely meager sampling of human motion, so the resulting joint transition graphs are far from being connected, which means that some joint orientations are just not reachable from others. Even when the graphs *are* connected, the search may have to wander all over the graph to find a path between two given vertices. If the corpus were large and rich, the graphs would be highly connected, which would give the search algorithms more leeway. In the existing corpora, however, the paucity of edges constrains the search to very narrow (and long) paths that can translate to stilted, idiosyncratic movement sequences. This is an unavoidable problem in this application, unfortunately; the dance world has not yet embraced the notion of computer animation, so the availability of animated dances is quite limited, and motion-capture studios are expensive to set up and run. The third interesting failure mode arises from the greedy search strategy, which creates "inefficiencies" in the interpolation sequences—places where the dancer appears to be headed towards the goal state, but then moves away. For example, one of the interpolation goals in figure 3 is to change the figure's facing from left to right. By the fourth frame, the dancer has turned to the right, but in the fifth frame s/he has turned back to the left again, which is part of what necessitates the relévé sequence between frames $\boxed{6}$ and $\boxed{7}$. Finally, note that some search strategies—e.g., always taking the highest-probability branch—can be a significant source of cliché.

The primary motivation for the development of these methods was our work on a mathematical technique[4] that automatically creates variations on predefined motion sequences—an idea that was inspired by a similar scheme[5, 6] that uses a related procedure to generate *musical* variations. This approach uses the mathematics of chaos to shuffle a predefined movement sequence by "wrapping" that sequence around a chaotic attractor. This establishes a symbolic dynamics that links the movement progression and the attractor geometry, which one can then use to generate variations on that original piece. Variations generated in this manner, whether musical or choreographic, are both aesthetically pleasing and strikingly reminiscent of the original sequences. The stretching and folding of the chaotic dynamics guarantee that the ordering of the pitches or movements in the variation is different from the original sequence; at the same time, the fixed geometry of the attractor ensures that a chaotic variation of Bach's Prelude in C Major or of a short Balanchine ballet sequence are related to the original piece in a sense reminiscent of the classic "variation on a theme." Broadly speaking, the chaotic variations resemble the originals with some shuffling of coherent subsequences. This is the primary source of the stylistic originality of the chaotic variation scheme — in fact, this type of subsequence shuffling is a well-established creative mechanism in modern choreography. One problem with any choreographic technique, automated or not, that involves subsequence reordering, however, is that the transitions at the subsequence boundaries can be quite jarring, and the interpolation algorithms covered in this paper can smooth these kinds of transitions in a manner that is both kinesiologically and stylistically consistent.

The "goal" of choreography is aesthetic appeal, so it is difficult to analyze the results of this work using standard scientific criteria[5]. However, there are some standard rules, procedures, and patterns in certain dance and martial arts genres; as described elsewhere[12], analyses based on these criteria suggest that MotionMind's sequences are indeed stylistically consonant. Another interesting way to evaluate these results is to construct a Turing test: say, ten sequences generated by a human choreographer and ten MotionMind sequences, in randomized order. We have put together such a test and administered it to roughly 100 people. The results are mixed; most of MotionMind's sequences are indistinguishable from human-generated ones, but a few are awkward in an artificial and recognizeable way. This, in turn, brought out another interesting variable; students who are majoring in dance found this awkwardness esthetically appealing, while computer science majors did not.

By applying techniques from statistics, graph theory, and heuristic search, the corpus-based interpolation methods described in this paper automatically construct interpolation sequences that move from one specified body posture to another in a *physically and stylistically coherent* fashion. Though our objective in doing this was to tailor generic strategies for a specific high-dimensional search problem to an unusual and demanding domain, the results could certainly be extended to other domains where the genre of sequence is important, such as speech recognition (e.g., filling in missing parts of a signal) or text. Finally, the implementation of these algorithms allows for arbitrary body topologies, so MotionMind is by no means limited to *human* motion sequences—though one would, of course, have to adapt the quaternion-based symbol set and the influence diagram to the topology of the limbs and joints that are involved.

[5]The very notion of objective, quantifyable evaluation elicited much consternation and mirth—along with some offense—from our dance colleagues.

References

[1] `www.cs.colorado.edu/~lizb/chaotic-dance.html`.

[2] J. Birringer. *Media and Peformance: Along the Border*. The Johns Hopkins University Press, 1998.

[3] E. Bradley, D. Capps, and A. Rubin. Computers and choreography. In *International Conference on Dance and Technology*, Tempe, AZ, 1999.

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

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

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

[7] J. Dunning. How to tell the computer from the dance. *New York Times*, February 1843.

[8] W. Hamilton. On a new species of imaginary quantities connected with a theory of quaternions. *Proceedings of the Royal Irish Academy*, 2:424–434, 1843.

[9] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *Proceedings of SIGGRAPH*, 1995.

[10] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[11] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

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

[13] W. L. Wooten. *Simulation of Leaping, Tumbling, Landing, and Balancing*. PhD thesis, Georgia Institute of Technology, 1998.