A Different Kind of Physics
Interactive evolution of expressive dancers and choreography

Kate Compton, Michael Mateas
Expressive Intelligence Studio
University of California, Santa Cruz
kcompton,mmateas@soe.ucsc.edu

Abstract

In this paper, we show how force-directed animation rules can be combined with spring-mass graphs to generate expressive motion. We present several games and prototypes demonstrating ways that this simple system can be used in games and computational creativity, to represent abstract creatures, kittens, and even evocative human actors, and to enable novel forms of interaction. Our final system evolves both ragdoll dancers and procedural choreography for them in a human-guided evolutionary algorithm. The choreography is a set of rules, forces applied to each node in response to musical variation, and this system can create a variety of interesting dancers and dances that respond to music.

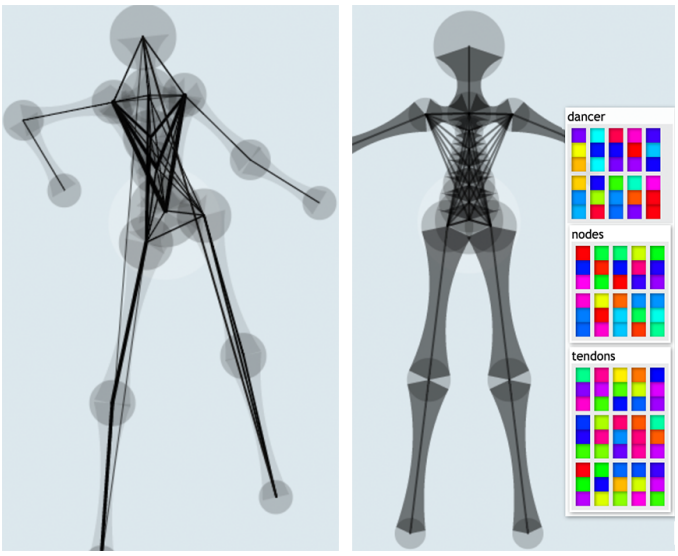

Figure 1: The ragdoll when moving and at rest. A network of springs connecting the spine to the shoulders and the hips to the spine create a sturdy and flexible tensegrity torso. Colorful DNA determines its form and movement.

Introduction

Dance has many purposes and many meanings, but one such purpose can be the expression of emotion through gestures. A study done with professional dancers (Sawada, Suda, and Ishii 2003) recorded their kinematic data to create a mapping from the emotion (anger, sadness, joy) with their arm movement qualities (speed, force, directedness). When novice viewers were asked to identify emotion in the choreographed movements, their perception of the emotion being expressed correlated with the qualities in the way that had been calculated from the original kinematic data.

It should not be surprising that choreography (even of a single body part) can express emotion: the famous Heider and Simmel animation experiment (Heider and Simmel 1944) demonstrated that an animated film can express very strong characters with only abstract shapes and motion, leading study participants to overwhelmingly label a triangle as "Aggressive, warlike, belligerent, pugnacious, quarrelsome, troublesome, mean, angry, bad-tempered". Interestingly, Heider and Simmel identify *rules and forces* (collisions, transfers of motion, movements towards or away from other 'actors') as the creators of meaning, rather than specific individual motions.

Expressiveness is an artistic quality of an animation, or of the affordances of a system, thus is hard to define. Animators, and those who make animation with code, agree that it comes from the *texture* (Perlin 1995) and *character* of the movement, not its specifics. Disney's famous Rules of Animation (Thomas, Johnston, and Thomas 1995) provide formal guidelines for expressive animation. They note that mimetic realism is *not* necessary or desirable; exaggerated and stylistic movement feels more 'real' than reality. One way to encode texture and character into movement is to model animation as rules rather than positions.

We draw inspiration from dance, another form of expressive movement. How is dance 'modeled'? Some choreography notations have specific notations for individual, discrete movements of a whole body, or of a single body part. Dances described with these languages are predetermined units of meaning strung together, recorded like words in a sentence. For groups of dancers, spatial relationships between dancers is of more significance, so "track drawings" show the movements of dancers through space. (Guest 1998)

Each form of dance notation balances *control* with *emergence*. As no notation can include all information, each system encodes some specific aspects of a dance, while allowing others emerge from the *rules*, or to be invented by the

reader/dancer if not specified. In this paper, we show a system, Teuhana, to evolve ragdolls and choreography for them.

Our representation of dance, like the notation systems of real choreographers, is designed for the specific affordances of our dancers. Our dancers are modeled as an object of springs and particle masses, a way of representing a figure that is less constrained, but with more potential for expressive rules than previous systems of animation. The animation, including choreography, is defined by force rules applied to those objects. This idea was developed and explored through the systems in Prior Work, and culminated in the Teuhana system.

Related Work

Inverse kinematics and procedural animation

When animators animate a model for games or film, they usually are specifying a *very specific* motion, to be played back perfectly on a pre-defined body. This form of animation models movement as a succession of keyframed positions and transition curves. When positions are unspecified, inverse kinematics (IK) solves them mathematically. This system has many advantages, it is predictable, deterministic, and any constraints can be solved with rigid dynamics or inverse kinematics.

Compare this form of animation (the overwhelmingly dominant paradigm in entertainment animation) with dance notation systems, which are, after all, just a different way of describing motion. If dance notation must balance between control and emergence, keyframe animation chooses *complete control* with no emergence. Which forms of animation chooses emergence instead?

Surprise and unpredictability are usually not desirable in professional animation, but interactive motion requires real-time animation generation. Ragdoll physics, the most common method of procedural animation for humanoid figures, models bodies as connected linkages of fixed length struts, often with constraints controlling the angles that limbs can move (Glimberg and Engel 2007).

Force-directed animation

A different kind of animation, *steering* specifies movement not by the resulting final positions, but the *forces* applied to create it. Steering can be applied as imaginary forces to the agents in the simulation (Reynolds 1987), or by simulating real physical forces from wheels and momentum (Braitenberg 1986). Both boids and Braitenberg vehicles are cybernetic systems that read information about the local environment (neighbor distance and alignment for boids, light or food for the vehicles) and turn that information into locally applied force. In these systems, *character* and emotion emerge through the forces applied.

Evolving Dance

The many kinds of dance notations are each designed to capture the most meaningful parts of a specific type of dance: floor movement for court dances, arm and leg positions for ballet, body torsion for modern dance (Guest 1998). Similarly, many systems have been designed to procedurally generate choreography, but each has its own focus and definition of what dance *is*. If the system uses evolutionary algorithms, it must encode its definition in both the representation of the dance and the heuristic for evaluating it.

Some dance can be represented as a sequence of static gestures. BharataNatyam dance is a series of precisely executed body postures(Jadhav, Joshi, and Pawar 2012), making a clear genetic representation possible, and is evaluated with the distance from the previous body posture, to keep all the limbs in motion, and distance to known good dances, to be both novel and feasible. Scuddle uses a similar model to describe modern dance with gestural curves showing body tension, optimizing for distance from the human choreographer's previous tendencies to provide novel inspiration (Carlson, Schiphorst, and Pasquier 2011). The work of Eisenmann et al expands the definition of 'choreography' as any coordinated group movement, and uses that concept to model dance as the top-down movement of *many* bodies through space, with Boids-like reactions to each others proximity. (Eisenmann et al. 2011).

Dubbin and Stanley generate *music responsive* dances, by evolving an Artificial Neural Network to turn analyzed music information into joint rotations for a body (Dubbin and Stanley 2010), the only one of the papers cited that uses rhythm and music as input for the dance.

Within a carefully constrained space of particular representations of dance, especially when guided by a human and not an abstractly calculated heuristic, these systems are able to create a range of possible dances from individual movements.

Making Expressive Tensegrity Objects

Tensegrity Ragdolls

Tensegrity, short for "tensile integrity" is an engineering method for creating structures of struts and cables (Pugh 1976). Tensegrity classically uses rigid struts and inelastic tendons (such as steel cable), but many practical tensegrity structures use elastic tendons or springlike flexible struts. With flexible or actuated components, a single rigid tensegrity structure can become a whole possibility space of movements and configurations, depending on the forces that are applied to it.

In an autotelic simulation such as a game, we can explore different physics tunings and tendon behaviors with no regard for real world feasibility. A similar technique, spring-mass diagrams, is used to simulate soft tissue in animation (Lee, Terzopoulos, and Waters 1995), and is usually optimized for "realism", but we can ignore realism in favor of allowing (and encouraging) tunings that create expressive non-realistic motion.

We used this concept to build a 'tensegrity' ragdoll, a humanoid figure made of springs connecting weighted nodes (Fig. 1). We also limit ourselves to two dimensions for drawing purposes (though we think that this method will generalize to 3D as well).

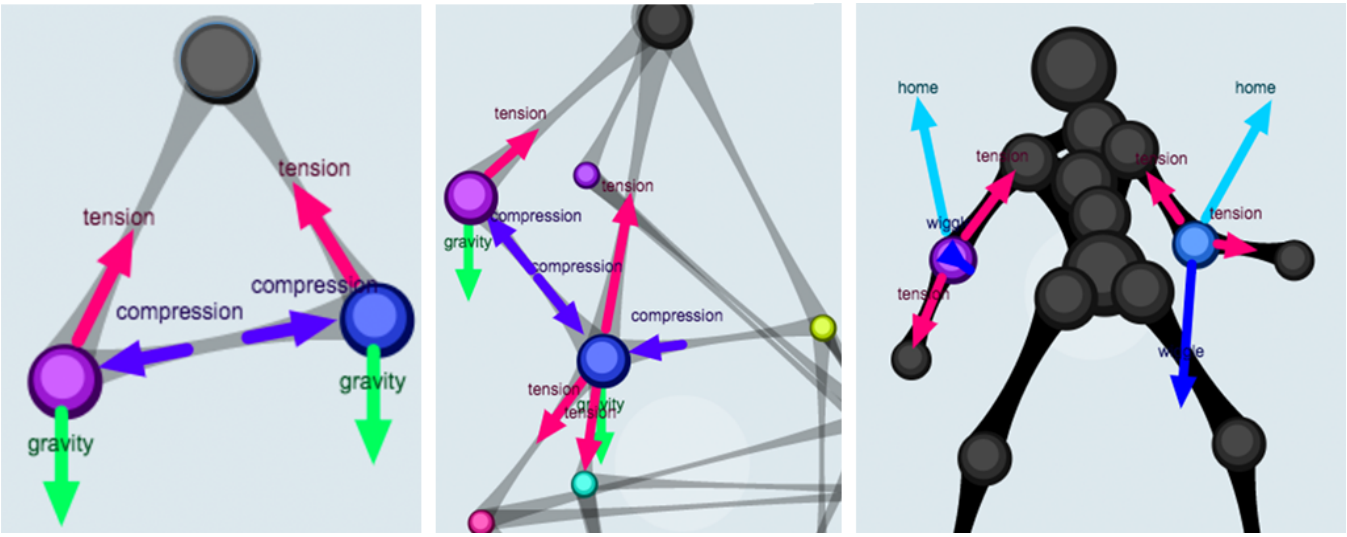

Figure 2: Applying forces to the tensegrity systems in our simulation. Left: tension, compression, and gravity forces create a stable form. Center: these same forces create a much more complex form, and pull it into a stable position. Right: the tensegrity forces create a stable, yet flexible, humanoid form. When additional forces are applied (*wiggle* to add motion, *home* to return to the original position) the doll can dance 'enthusiastically', yet remain in a recognizably human shape. (note: all labels and arrows in these diagrams are actually in-game; this makes debugging very easy)

A tensegrity ragdoll captures a different kind of body movement that rigid-body ragdolls lack. Squash and stretch, arcing motion, exaggeration, and secondary action (as feedback from the spring travels through the rest of the body) all emerge naturally from this system, and are also four of the Disney principles for making appealing animation (Thomas, Johnston, and Thomas 1995).

Animating tensegrity objects

These tensegrity objects are easy to program, and the forces can be easily visualized, especially in 2D (Fig. 2)

- Maintain a graph structure of nodes and edges
- Model each node as a particle, with a mass, position, velocity, and the current force being applied
- Each frame:
  1. Clear all the forces on the particles
  2. Calculate all the forces like gravity, repulsion, attraction, etc. on each particle
  3. For each edge, calculate its offset from the ideal length, and apply a force to the particles that it connects, to draw them together or push them apart
  4. Sum up the forces on each particle, and add the particle's force (divided by its mass) to its velocity, and add the velocity to the position
  5. (optional) Perform some additional easing of the edges by physically moving the particles at the ends of the edges (bypassing applying force)
  6. (optional) Create 'drag' by multiplying the velocity by (1 - drag) to make the particles feel sticky or slow

Note that there is no explicit collision detection or constraint satisfaction at work here. It does not accurately model physics: the process is tuned for expressiveness. The "drag" and "easing" steps are used to tune the *texture* of the motion, and only incidentally have physics implications. Nor is this a *replacement* for traditional ragdolls or inverse kinematics in traditional game genres; instead it enables *novel* forms of gameplay, as seen in our games **Falling for You** and **Teuhana**.

Prior Work

Scribbles

In **Scribbles**, made in 2004, we explored with expressive tensegrity objects as part of a living drawing tool experiment. As the user draws a segmented curve on the screen, the system independently generates tendons between the nodes, creating a graph of nodes and edges. These tendons would contract and release, causing the "bones" of the original user-sketched curve to twitch and move in a lifelike manner, often crawling offscreen or spiraling around like an out-of-control siphonophore. Even these very abstract creatures had distinct personalities that emerged from the specific morphology of their generative forms.

Falling for You

Falling For You (Fig. 3) uses similar tensegrity forms to create humanoid ragdolls (and balloons) which would respond to simplified physical forces like gravity and attraction. Their nodes have mass, density, and weight which affected how they responded to physical force. Gravity caused most nodes to be pulled downward, but some, like the balloons which had negative weight, were pulled up instead.

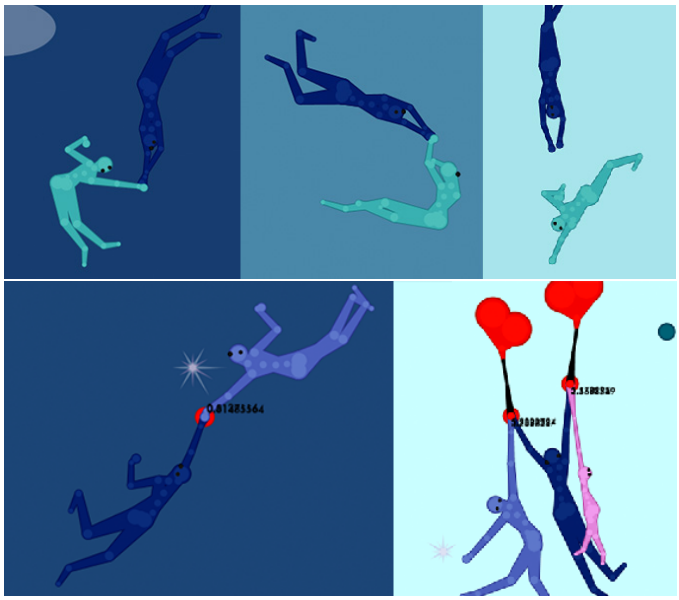

Figure 3: Common emotional vignettes during Falling for You (2011) gameplay. Top: one doll 'loves' another (its hand is attracted to the hand of the other) but the other doll in indifferent. As the turquoise doll drifts away unaffected, the blue doll stretches out, seemingly in desperate longing, drawn forward by its hand. Bottom: mutual attraction pulls two dolls together by their hands. The intersection of their hands generates a 'love balloon', and later a baby with another love balloon. The whole family is drawn upward, bound together by their hands.

The nodes representing the hands are subject to *emotional* forces as well. "Attraction" was modeled as a directional pairwise boolean: if the owner of the hand was attracted to the owner of another nearby hand, that hand received a strong attraction force, pulling it hard towards the hand of its desired partner. Because the bodies were fully connected with stretchy tendons, as the hand pulled towards a goal, the whole body would be pulled progressively towards it as well, starting from the arms, shoulders and heads, then the hips, and finally the feet, as each internal spring was stretched to the point of applying a force.

With this simple rule, ragdolls who otherwise float aimlessly stretch their hands longingly to the hands of their beloved, who would either stretch their hands back in reciprocation, or drift obliviously away if indifferent. When two mutually-attracted hands intersect, they link and form a heart-shaped balloon, which pulls them upwards, stopping their downward fall, and turning the two ragdolls into a single tensegrity object held together by the metonymic union of the hands.

A strong enough force could pull the hands apart, and the two ragdolls would drift apart again, though the physics were tuned such that this breakup would not happen without external interaction from the player.

We often describe emotions and relationships as physical forces: we were drawn together, I was attracted to you, we drifted away, we were torn apart, I pulled away from you. You were magnetic, irresistible or perhaps you were repulsive, repellant. As anticipated by Heider-Simmel experiment, modeling these metaphorical forces as *actual physical* forces creates readable emotional animation.

Falling for You allowed interaction via the left and right arrow keys: holding either direction key applies a force uniformly to the all nodes of the main doll. The ragdolls also act autonomously, so the game could be played passively without any interaction at all, and the main ragdoll (the one who followed by the camera) will drift, grabbing partners and forming families without any external input, though only the player's force is strong enough to break up established relationships.

Combining user-provided forces and internally-generated forces suggests a new way of balancing player agency with diegetic character desires. As in a good Aristotelian drama, the actor has their own motivations, but those motivations are overwhelmed or redirected by higher forces that cause them to work against their natural instincts. The character's desire is not *overwritten* by the external forces: we still see them struggling like a puppet against its strings, a struggle which creates drama. When the player pulls their doll away from a relationship, they can feel the force of the springs pulling tight: the relationship is exerting *force* against being broken.

Prom Week (McCoy et al. 2011) and the Sims (Maxis 2000) similarly give their little characters some agency and self-direction. The Sims characters will follow their desires to choose from the local space of possibilities, but when the player redirects them, they cannot express a resistive force, only a binary refusal or compliance. Prom Week models character choices as purchasable actions, which allows more gradation of resistance ("social physics"). A character can perform their preferred action for free, but getting a shy character to flirt, the jock to ask the nerd to prom, or a happy girlfriend to break up with her boyfriend, is more expensive to represent the narrative force needed to get the characters to perform against type. The resistance force is not modeled physically but numerically, but it can still be read as an expressive force.

Novel Interaction Affordances

When we think about the potential gameplay applications of these ragdolls, we should focus on what strange new interactions they can afford, rather than considering them as a replacement for current ragdoll technology. Falling for You used semi-indirect control of controlling forces to the ragdoll's nodes. Other potential indirect interactions could include interacting with the global forces like gravity, tides, or wind to see their affect on the ragdolls.

Conversely, more direct control could be achieved by letting the user directly drag the nodes. We prototyped this with a variant of Falling for You called **Oh No, Kitten Avalanche!** in which the ragdolls were replaced by gelatinous kittens that had to be flung off-screen by dragging any of their nodes. This had some very appealing affordances: the choice of node (foot, spine, head, tail) affected how

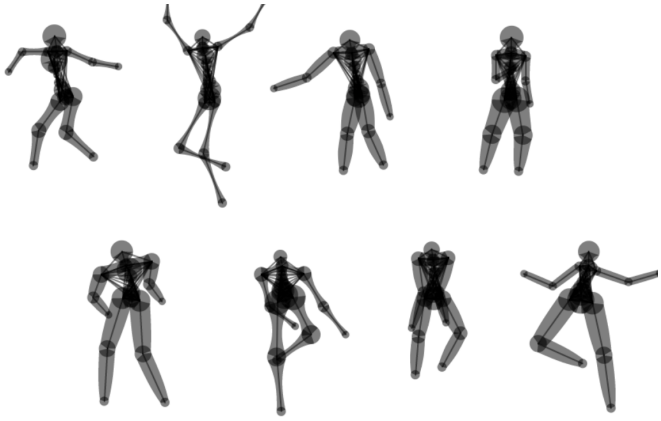

Figure 4: Many morphologies in a single Teuhana generation.

much the kitten's natural tensegrity could resist the movement (an issue familiar to real-world cat owners).

Since our force-directed animation is dynamically generated, we can respond to *any* kind of input that we can translate into in-game nodes or forces. We have hooked up simulated tensegrity graphs to make Kinect poi, pendulums dangling from fingertips with the LeapMotion, and a beard made of springs. These are all preliminary experiments, with no antagonistic or winnable gameplay yet, but they demonstrate the potential for using these systems for intuitive and novel interaction.

Teuhana: Evolving Dancing Ragdolls

In Teuhana, we further explore these ideas by combining tensegrity ragdolls with a human directed genetic algorithm. The tendons in the ragdolls from Falling For You were hand-tuned, could we use human-directed genetic algorithms to do this tuning more quickly, and for a wider range of morphologies? Can we then use the *same process* to evolve animation controls?

Previous systems have used genetic algorithms to guide computational creativity systems to generate better artifacts (BoxCar 2D), (Genetic Algorithm Walkers). This technique has also made real-world tensegrity robots feasible (Paul, Valero-Cuevas, and Lipson 2006) .

Automatic evolution works if there is a computable heuristic, but often the real heuristic of a computational creativity system is "it looks good", or "it looks good *to me*". Human-guided evolution, or Interactive Evolutionary Computation (IEC) (Takagi 2001) allows human intuition to provide this heuristic. Users of the systems PicBreeder (Secretan et al. 2011) quickly navigate a space of complex generated artifacts by picking the ones they like, and can guide the evolutionary process to the most pleasing or interesting artifacts.

Evolving Form

An evolutionary algorithm needs three components: a phenotype that can be evaluated, a genotype that can be modified (via crossover or mutation), and a way to turn the latter into the former. In Teuhana's ragdolls, the phenotype is the a graph of springy tendons and nodes that makes up the body. The genotype is the information needed to create that graph.

We could have a representation of each node's radius, density, and position and each spring's ideal length, strength, and easing, but this would encompass many bad solutions. The art of a successful genetic algorithm is in carefully sculpting the possibility space, to encode 'good' aesthetics implicitly in the generative algorithm. We have found that having several higher-level aesthetic variables encoded in the genotype results in more controllable aesthetic outcomes, so part of the dancer's genotype encodes concepts like "stiffness", "bulkiness", "lankiness", and "torso triangularity." (see appendix for full genome)

Evolving Movement

At a glance, it is easy to see different shapes of dancers, and pick out which ones to evolve further, as seen in Figure 4). But a dancer that looks good in a still image may move in very unpredictable ways once forces are applied. Simulating the dancers allows the user to easily scan the dancers for good movement.

Responding to Stimula: Dance Responders As shown in Related and Prior Work, forces acting on a body can create expressive movement. From personal experience with swing and ballroom dances, we know this is also a good model of dance, so Teuhana represents dance as a set of forces rules, generated by the music, acting on the ragdoll. We use **dancer.js**, a free realtime music analysis library, to break any song we load into its component frequencies. A dance is represented by a set of *dance responders*, each of which is attached to some node in the body, and listening for a particular frequency. When the responder hears its note, it actuates, applying some force to the node. So as the music plays, different dance responds kick at different moments, at different parts of the body.

Species of Dance Responders The *genotype* of a dance is the DNA of its dance responders: which node they occupy, what note they hear, and what they do when they hear that note (see appendix. The last attribute is dependent on their 'species': rather than trying to get good moves out of a universal reaction rule, we implemented several kinds of rules that express common real-world dance moves, while still leaving a lot of room for genetic variants (Fig. 6).

Future Work

The obvious next work to be done is to perform formal user testing. Are users able to direct the evolution of the dancers and choreographic rules to create creative new dances? Do these dances have perceivably distinct 'character', and does that character persist when the dance is performed by new dancers?

Additional work will extend the character morphology to include accessories that traditionally create secondary motion in dance: tassels, poi, and skirts, and less traditional accessories like multiple arms. We hope to direct the forces

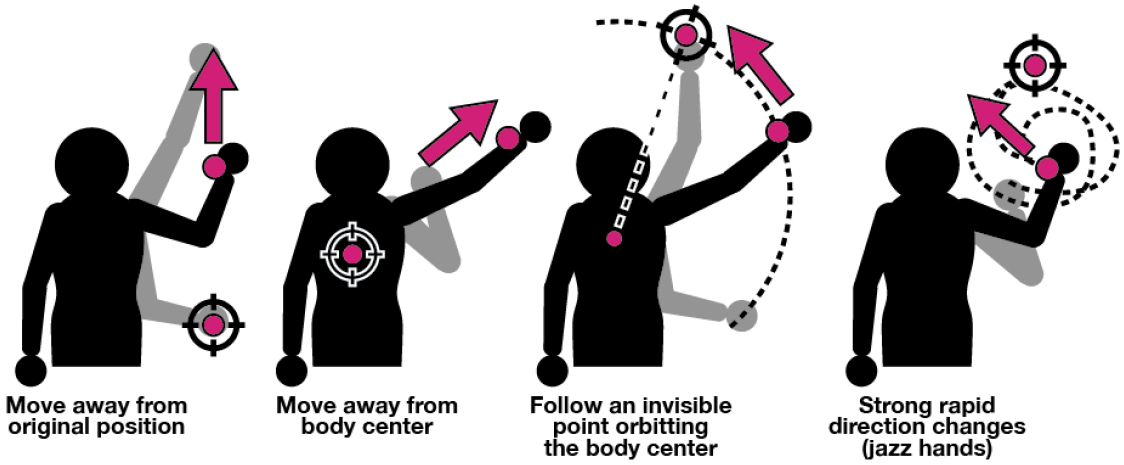

Figure 5: Several "species" of dance responders. Each applies a different force to its node according to its own dance responder DNA **and** the local situation of the node that it inhabits. A heavy node on a stiff tendon, like a head, will not show the force in the same way that a light flexible node like a hand would.

with motion-control systems to provide more user control than simply selecting the descendants of the evolution.

References

BoxCar 2D. *http://boxcar2d.com/*. Accessed: 2015-05-04.

Braitenberg, V. 1986. *Vehicles: Experiments in synthetic psychology*. MIT press.

Carlson, K.; Schiphorst, T.; and Pasquier, P. 2011. Scuddle: Generating movement catalysts for computer-aided choreography. In *Proceedings of the Second International Conference on Computational Creativity*, 123–128.

Dubbin, G. A., and Stanley, K. O. 2010. Learning to dance through interactive evolution. In *Applications of Evolutionary Computation*. Springer. 331–340.

Eisenmann, J.; Schroeder, B.; Lewis, M.; and Parent, R. 2011. Creating choreography with interactive evolutionary algorithms. In *Applications of Evolutionary Computation*. Springer. 293–302.

Genetic Algorithm Walkers. *http://rednuht.org/genetic_walkers/*. Accessed: 2015-05-04.

Glimberg, S., and Engel, M. 2007. Comparison of ragdoll methods.

Guest, A. H. 1998. *Choreo-graphics: a comparison of dance notation systems from the fifteenth century to the present*. Psychology Press.

Heider, F., and Simmel, M. 1944. An experimental study of apparent behavior. *The American Journal of Psychology* 243–259.

Jadhav, S.; Joshi, M.; and Pawar, J. 2012. Art to smart: an evolutionary computational model for bharatanatyam choreography. In *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*, 384–389. IEEE.

Lee, Y.; Terzopoulos, D.; and Waters, K. 1995. Realistic modeling for facial animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 55–62. ACM.

Maxis. 2000. The sims. *Electronic Arts*.

McCoy, J.; Treanor, M.; Samuel, B.; Mateas, M.; and Wardrip-Fruin, N. 2011. Prom week: social physics as gameplay. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, 319–321. ACM.

Paul, C.; Valero-Cuevas, F. J.; and Lipson, H. 2006. Design and control of tensegrity robots for locomotion. *Robotics, IEEE Transactions on* 22(5):944–957.

Perlin, K. 1995. Real time responsive animation with personality. *Visualization and Computer Graphics, IEEE Transactions on* 1(1):5–15.

Pugh, A. 1976. *An introduction to tensegrity*. Univ of California Press.

Reynolds, C. W. 1987. Flocks, herds and schools: A distributed behavioral model. *ACM Siggraph Computer Graphics* 21(4):25–34.

Sawada, M.; Suda, K.; and Ishii, M. 2003. Expression of emotions in dance: Relation between arm movement characteristics and emotion. *Perceptual and motor skills* 97(3):697–708.

Secretan, J.; Beato, N.; D'Ambrosio, D. B.; Rodriguez, A.; Campbell, A.; Folsom-Kovarik, J. T.; and Stanley, K. O. 2011. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation* 19(3):373–403.

Takagi, H. 2001. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE* 89(9):1275–1296.

Thomas, F.; Johnston, O.; and Thomas, F. 1995. *The illusion of life: Disney animation*. Hyperion New York.

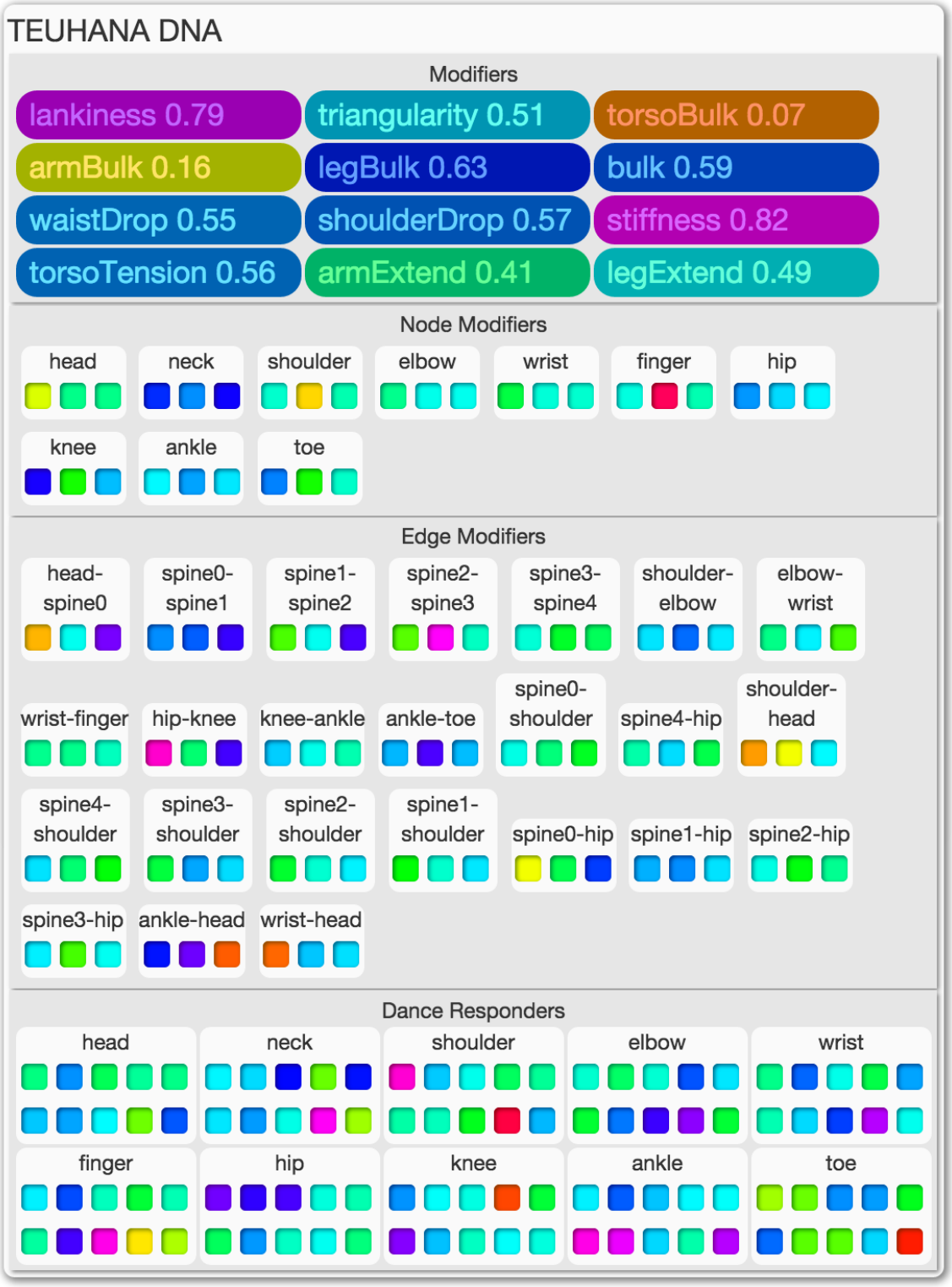

Figure 6: Each square (and the top values, are floating point values [0, 1]. Main body modifiers (lankiness, etc), though controlled by only a single value, are complex deformations of multiple nodes and edges, allowing a holistic specification of body types. Node modifiers can further tune the radius, density, and position that were calculated from the main modifiers, and so do the Edge modifiers for ideal length, strength, and easing. Dance responders' values set the 'species' of responder (by buckets, with 0 being none), the force strength, the music pitch listened for, the music pitch width, and one custom tuning value for each responder. Each node may have up to two responders.