Sticky Feet: Evolution in a Multi-Creature Physical Simulation

Greg Turk

School of Interactive Computing
Georgia Institute of Technology, Atlanta, GA 30332
turk@cc.gatech.edu

Abstract

We demonstrate artificial evolution in a system that combines physical simulation with competition between creatures. The simulated creatures are constructed using point masses that are connected by oscillating springs. The creatures pull themselves across their 2D environment by varying the amount of friction at different point masses, giving them sticky feet. Creatures combat one another, and the victor of such an encounter earns the right to reproduce, possibly with mutation. Rather than testing one individual against another in pairs, as many as 100 creatures move and interact with each other in the same 2D environment. Over time, the initial creatures are replaced by new creatures that are more agile and better at combating others. The evolved creatures from such simulations exhibit a wide array of body plans, locomotion styles, and interaction behaviors.

Introduction

The animal kingdom displays an astonishing variety of creature body plans, methods of locomotion, and styles of interactions between individuals. The engine that produces this seemingly endless array of forms and behaviors is Darwinian evolution. One of the goals of Artificial Life is to demonstrate that a similar degree of richness can be produced by unguided evolution in a computer-simulated world. Success in creating rich simulated worlds can inform our understanding of real-world evolution and may also be a valuable teaching tool, allowing students to witness a process that is slow in nature.

Our research is inspired by prior work in Artificial Life, and in particular, by simulated creature evolution through the use of physical simulation. A particular goal of our work is to create a single environment in which many creatures interact with one another, reproduce and evolve. We wish to simulate as many creatures at one time as possible in order to have a sufficiently large population in the environment. This led us to seek the most simple virtual bodies that would still exhibit a variety of shapes and behaviors. We selected point masses that are connected by springs as our representation of a creature's body. Each virtual spring may change its rest length in a cyclic manner. By changing the friction on either end of such an oscillating spring, a creature uses its sticky feet to pull itself through the environment. The creatures live in a 2D world that has no gravity and no ground plane, so the creatures may crawl in any direction. Although this is a simple virtual physics model, the evolved creatures based on this model show a considerable variety in their body shapes and motions.

In our experiments, just one small moving creature is introduced into the virtual environment. This ancestral creature is initially surrounded by stationary creatures that cannot defend themselves, and these act as food for moving creatures. The lone moving creature "eats" the stationary ones, and it replicates after doing so. After a while, many of these small moving creatures are crawling through the environment. An occasional mutation occurs during replication, and the environment is soon filled with a variety of creature types. Some of these new creatures are more successful at combat and reproduction, and eventually the ancestral creature is supplanted by its more agile descendants. Different simulation runs have exhibited a wide variety of successful creature body plans and modes of locomotion.

The remainder of the paper is divided as follows. After discussing related work, we then describe the creature bodies and the physics simulator in detail. Next, we describe the mechanism by which creatures interact and reproduce, followed by a description of the allowed creature mutations. We then present the results of our simulation runs, followed by a discussion of future work.

Related Work

There are two main lines of research that are closely related to our own, and we review the research in each area in turn. The first area of research that is related to ours is simulated physics for creature locomotion. In 1993, two research groups demonstrated the evolution of creature locomotion that is based on simple virtual physics. Van de Panne and Fiume constructed creatures from rigid segments in 2D that use linear and angular actuators in order to move. They use simulated annealing to search for control networks that lead to efficient locomotion, such as walking and jumping, for a given creature body (Van de Panne and Fiume, 1993). Ngo and Marks simulate 2D creatures that are composed of rigid linear elements and creature-controlled angular joints, and they use a genetic algorithm to evolve more efficient locomotion. Their approach produces a variety of

walking, crawling and jumping creatures (Ngo and Marks, 1993). These initial approaches were used to develop controllers for a fixed creature body plan. Sims extended this work by evolving the creature bodies as well as their controllers using a genetic algorithm (Sims, 1994b). His virtual creatures are entirely 3D, and they are composed of blocks that are connected with joints that are controlled by the creature. This approach produces compelling examples of creature motion for creatures that walk, jump and swim. Komosinski and Rotaru-Varga use a physical simulator to investigate the effectiveness of different genotype encodings to explore the space of locomotion strategies (Komosinski and Rotaru-Varga, 2001). Lipson and Pollack use physical simulation to evolve crawling creatures made of rods that they then manufacture using rapid prototyping (Lipson and Pollack, 2000). Taylor and Massey give an excellent review of much of the research that has been done using physical simulation (Taylor and Massey, 2001).

The second area related to our research is the study of virtual creature interactions. In most of this research, the creature's bodies are simple and fixed, and the creature motions are the result of simple steering. Many of the Artificial Life models for creatures that sense and move have been inspired by the essays of Braitenberg on vehicles whose behaviors are governed by simple neural circuitry (Braitenberg, 1984). Yeager's PolyWorld simulator consists of creatures with a simple body, but with complex neural circuitry to control behavior (Yaeger, 1994). A large number of PolyWorld creatures compete for food, mate, and reproduce in a single environment that allows any creature to interact with each other. Reynolds uses the game of tag to co-evolve creatures that are good at pursuit and evasion (Reynolds, 1994). He evolves more skilled creatures using a genetic algorithm, and evaluates a fitness function by playing creatures against one another in pairs. Miller and Cliff argue that co-evolving pursuit and evasion strategies is an important topic for robotics and other applications (Miller and Cliff, 1994). Ventrella demonstrates evolution of swimming creatures in a simulated environment in which many creatures interact (Ventrella, 1996). His swimmers compete for food and then mate in order to reproduce. Miconi simulates a micro-world of block creatures that inflict damage on one another and that reproduce based on their health status (Miconi, 2008). The systems of Ventrella and Miconi are similar to our own in combining physical simulation with a multi-creature environment that fosters between-creature competition.

Creature Locomotion

A main goal of our research is to create a simulated environment in which many creatures can interact with one another. Because we wanted to simulate many creatures at once, we use an artificial physics that is computationally inexpensive, yet still capable of creating a wide range of motions. Our creatures are made of a collection of point-masses that are

connected by segments that are each linear springs. Each of these segments can be directed to change its rest length in a periodic manner, which causes the segment to oscillate in length. These creatures live on a 2D plane that has no gravity, so that there is no preferred orientation. This means that a given creature can approach another creature from any direction.

The equations that govern the motion of these creatures are those of a damped spring. For a segment that connects points with positions P_i, P_j and velocities V_i, V_j, the spring force F_i^{spring} acting on mass i is given by:

$$
\begin{aligned}
L &= P_i - P_j \\
\dot{L} &= V_i - V_j \\
F_i^{spring} &= -(k_s(L - L_{rest}) + k_d \frac{(\dot{L} \cdot L)}{|L|}) \frac{L}{|L|}
\end{aligned}
$$

In the above equations, after (Baraff et al., 1999), the spring strength k_s and spring damping coefficient k_d are set to be the same across all springs of all creatures. L_{rest} is the rest length of a particular segment, and it can vary periodically. For a segment with an original length L_{seg}, oscillation amplitude a, frequency f, and phase p, the change to its rest length is given by:

$$
L_{rest} = L_{seg}(1 + a\sin(ft + 2\pi p))
$$

The most simple creature that can move is composed of two point masses that are connected by a single segment. If this segment oscillates in the absence of other external forces, the creature's center of mass will remain unchanged. In order for such a creature to move, this creature must have a way to gain traction. Miller solved the traction problem by used directional friction in order to simulate the motion of worms and snakes (Miller, 1988). Our simulated creatures gain traction in a similar manner, by periodically changing the coefficient of friction at the two endpoints of the spring in synchrony with the oscillation of the segment itself. That is, each point alternates being sticky or slippery. Figure 1 shows a one-segment creature where the leading point is on the top and the trailing point is on the bottom. When the segment is elongating, the lead point is frictionless (shown using a smaller radius) and the trailing point is given a high friction coefficient (shown with a larger radius). This pushes the creature in the direction of the leading point. When the segment is shortening, the leading point is sticky and the trailing point is allowed to slide, and this pulls the creature towards the leading point. This single-segment creature moves along much like an inchworm.

To modify the stickiness of a given point-mass i, a per-point friction force is calculated that is proportional to the point's velocity and a global friction coefficient k_f:
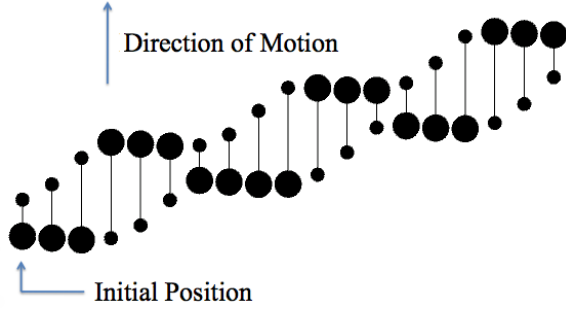
Figure 1: *A simple one-segment creature that moves up-wards. The creature's initial state is shown at the left, and subsequent positions in time are shown to the right of this. Points with high friction have a large radius, and the smaller points have lower friction.*

$$F_i^{friction} = -k_f f_i V_i$$

This per-point friction is modulated based on the phase of any spring segment that is attached to point i. We define a coefficient f_k for a given spring that is based on a per-spring friction magnitude m_k and the phase of the oscillation:

$$f_k = \begin{cases} -m_k & \text{if } \cos{(ft + 2\pi p)} \leq 0 \\ m_k & \text{if } \cos{(ft + 2\pi p)} > 0 \end{cases}$$

If a given spring k connects particles i and j, the friction coefficient f_k is *added* to particle i's friction accumulation f_i, and f_k is *subtracted* from particle j's friction accumulation f_j. Thus a spring will cause one of its particles to become more sticky and will cause the particle on its other end to be more inclined to slide. Once the friction accumulation f_i for a given particle has been modified by all of the attached springs, the value of f_i is then clamped to the range $[0, 1]$. Freely sliding particles have a value for f_i at or near zero, and points with larger values of f_i are "sticky", and resist motion.

Figure 2 shows a more complicated creature body, consisting of three point-masses and three segments that form a triangle. Assume that two of the segments oscillate out of phase with each other, so these sides of the creature lengthen and shorten alternately. Further assume that the third segment does not oscillate. If the friction magnitudes m of the two changing segments are the same, then the two sides take turns pushing the creature forward. Such a creature moves forward with a locomotion gate that looks like a waddle.

Our use of oscillating springs was partly inspired by the SodaPlay mass-spring simulator (Burton, 2007). Constructions in SodaPlay consist of point masses that are connected by springs. Any spring may be set to vary its length in a periodic manner, and constructions from such springs and

point masses move around in a 2D environment. Unlike our model, SodaPlay constructions live in an environment with gravity and a floor.

Competition and Sensing

Pursuit and evasion contests are among the most common types of creature interactions in nature. Predators chase their prey, and the prey try to evade capture. Creatures of the same species chase each other when they are competing for food or mates. Because of the real-world importance of these behaviors, several researchers have made convincing arguments in favor of studying pursuit and evasion in artificial simulations (Reynolds, 1994; Miller and Cliff, 1994). Our own work takes inspiration from this prior work, and the artificial evolution in our simulator is driven by the success that the creatures have in pursuing one another.

Because our simulated creatures are composed of multiple point masses and segments, we must define what it means for one creature to capture another. Each creature has one of its point masses designated as its *mouth*, and a different point-mass as its *heart*. One creature successfully captures another when the mouth of the first creature comes within a specified radius of the heart of the second creature. There are several consequences of this model of competition. First, it means that any creature may be the aggressor or the chased. Second, it is very unlikely that a pair of creatures simultaneously capture each other. Finally, it allows the morphology of a creature to be tailored to the nature of the mouth and the heart. For instance, a successful creature is likely to have its mouth placed forward relative to its direction of motion.

All of our virtual creatures live together in one large 2D world, and a typical population consists of 100 creatures. Creatures encounter each other as they crawl forward in this world. When one creature successfully captures another, the victor of the encounter is rewarded by being copied (reproduction), and the loser of the encounter is removed from the
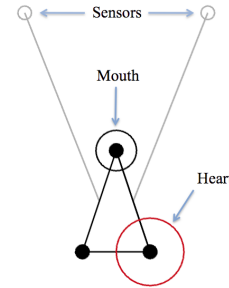

Figure 2: *A simple triangular creature with three point-masses and three segments. Two of the segments have sensors attached to them that extend in front of the creature. The mouth of this creature faces its direction of motion, and its heart is in a trailing position.*

simulation. In this manner, the creatures that are more successful at pursuit become more numerous, and the creatures that often lose such competitions are eventually eliminated from the population. In most cases, the winner of the competition is duplicated exactly, but on occasion there may be one or more mutations that occur during reproduction. In this way, new creature body plans and behaviors can emerge.

This mechanism of closely tying competition with reproduction is similar to a steady-state genetic algorithm, since exactly one member of the population is replaced at a given time. In our simulations, the fitness evaluation is the outcome of a single encounter. This is a departure from much of the prior work on artificial creature evolution, where the fitness function for a creature is usually determined by mini-tournaments between pairs of creatures (Reynolds, 1994; Miller and Cliff, 1994; Sims, 1994a). We believe that having all the creatures in one large environment is closer to modeling the real world than the alternative of mini-tournaments. In addition, placing all of the creatures in a single environment allows for a richer set of encounters. A creature has to select its own prey, and may change to another creature target mid-way through an attack. It is possible for one creature to be both the pursuer of a second creature, and to simultaneously be chased by a third creature.

In order for a creature to recognize the presence of another creature, each creature can modulate the motion of its segments based on the output of proximity sensors. More specifically, each segment can have one sensor that is tied to that particular segment. Thus a creature that is composed of three segments may have up to three sensors, one for each of its segments. Each sensor recognizes the presence of either a heart or a mouth of another creature. A sensor is defined by several attributes: its position relative to the segment, its sensing radius, what body part it senses (heart or mouth), its modulation strength, and the type of modulation that it uses to affect its segment.

A sensor has an all-or-nothing response, depending on whether another creature's heart or mouth is inside the sensor's radius. Each sensor has a modulation strength m that can be positive or negative. If a sensor is triggered, it changes the property of the oscillating segment that it is tied to in one of three ways. When triggered, the sensor modulation strength m may be added to the amplitude of the segment's length a, it may be added to the friction force f_k of the segment, or it may be added to the friction magnitude m_k of the segment. Thus a sensor may cause a spring to oscillate more or less, it may cause points to become more or less sticky, or it may alter which of the endpoints of a spring are sticky at a given time (possibly slowing or reversing the direction of motion).

Figure 2 shows a three-segment creature that has two sensors that are positioned forward of the creature's direction of motion. Assume that each of these sense the proximity of another creature's heart, and that upon doing so, this causes the magnitude of the spring oscillations to decrease. If the presence of another creature's heart sets off the right-hand sensor, this will cause the creature to be pushed forward more weakly on its right side. This makes the creature turn towards the creature that triggered its sensor. In this way, a simple creature can sense and pursue other creatures. This method of steering based on sensors and motors is in the spirit of Braitenberg's vehicles (Braitenberg, 1984).

Both proximity sensing and the determination of creature capture require testing whether one point is within a given radius of another point. In a naive implementation, testing whether each creature's mouth is near any other creature's heart requires $O(n^2)$ operations for n creatures. We speed up this test by first noting that each creature's heart has a fixed radius r. To rapidly determine mouth/heart proximity, we first create a grid of square cells with side lengths r that is superimposed on the 2D environment in which the creatures live. Each cell in this grid maintains a list of the hearts that fall within the cell at the current time-step. To test whether a creature mouth is near to any hearts, only nine cells need to be checked, namely the cell that the mouth is currently in and the eight neighboring cells. Testing whether a sensor is close to a mouth or a heart is similar, but in this case the cell size is given by the maximum radius of all sensors.

Creature Reproduction

When one creature captures another, it is rewarded by being replicated, possibly with mutation. In our simulations we used a mutation rate of 0.1, so that one out of ten creature replications occurs with mutation. This is a much higher mutation rate than is typically used in a genetic algorithm. Note that in a genetic algorithm, most of the variation is generated by crossing-over, and we do not have such a mechanism in our simulator. We also have a fairly high probability of multiple mutations during reproduction. If a creature is to be mutated, there is a probability of 0.4 that it will have a second mutation, 0.4^2 that it will have three mutations, 0.4^3 for four mutations, and so on. Mutations can be grouped into three categories: per-segment physical parameters, sensor parameters, and creature body shape.

Per-segment mutations alter parameters that are specific to a segment that is chosen at random. The possible mutated parameters are segment length, amplitude of oscillation, frequency of oscillation, phase of oscillation, and the magnitude of change that the segment uses to alter the friction of its endpoints.

A sensor mutation alters one of the parameters that guides the action of a creature. For most of these, a segment is chosen at random and the parameters of the segment's sensor is altered. Potential changes include the angle of the sensor relative to the segment's orientation, the distance of the sensor from the segment, the radius of the sensor, and the

sensor type (senses mouth or heart). There are three other mutations that change what the sensor modifies (segment length amplitude a, friction force f_k, or friction magnitude m_k). When one of these three mutations occurs, the sensor is switched to modifying a particular segment parameter, and a new sensor modulation strength m is chosen. A final type of behavioral change that can occur is the verbatim copying of all the sensor parameters from one segment to another. This mutation was designed in recognition of the fact that many advances in biological evolution occur due to duplication and divergence.

The final class of mutations are changes to the creature's body plan. One such mutation modifies the position of either the heart or the mouth at random. Another body mutation deletes a segment at random. This mutation is only deemed valid if deleting the segment would not separate the creature into disjoint components. Another mutation adds a segment that is attached to the other segments only at one end, producing a dangling segment. Note that such dangling segments can still contribute to a creature's motion. One mutation fuses two such dangling segments, and another connects two dangling ends with a new segment. Finally, one mutation attaches two new segments to an already existing segment in a manner that forms a new triangle.

When a creature is replicated, regardless of whether or not it is a mutation, the new creature is placed in the 2D environment at a random position and orientation. The placement algorithm makes sure that the creature's segments do not overlap with any already existing creatures. This is done by repeated attempts to place the new creature at random locations in a non-overlapping manner. The placement algorithm can in rare cases terminate unsuccessfully after a fixed number of placement attempts, and the maximum number of placement trials is set to 40 in our simulations. Such placement failures are an indication that the creatures are developing substantially larger bodies, and in such cases the population size gradually decreases (through placement failures) to accommodate this change in creature body size.

Simulation Results

We ran three classes of simulations, namely lone ancestor runs, between-generation contests of evolved creatures, and a tournament across creatures from many different simulation runs. We report on each of these kinds of simulations in turn. (Video of these results can be found at `http://www.cc.gatech.edu/~turk/stickyfeet/`)

All of the lone ancestor simulations were conducted using the same initial conditions, with the only difference between runs being differences in the random number seeds. In each of these runs, the simulation begins with 100 creatures. All but one of these initial creatures are motionless one-segment creatures with hearts but without mouths. By design, these static creatures cannot win an encounter with another crea-

ture. In effect, these 99 motionless creatures act as a potential food source for other creatures. The one moving creature had a one-segment body, and it moves by changes to its segment length and by synchronized changes in friction to its two point-masses. This forward motion is illustrated in Figure 1. The forward point-mass of this creature is its mouth, and the back point is its heart. Sensors are not shown in this and later figures to avoid visual clutter.
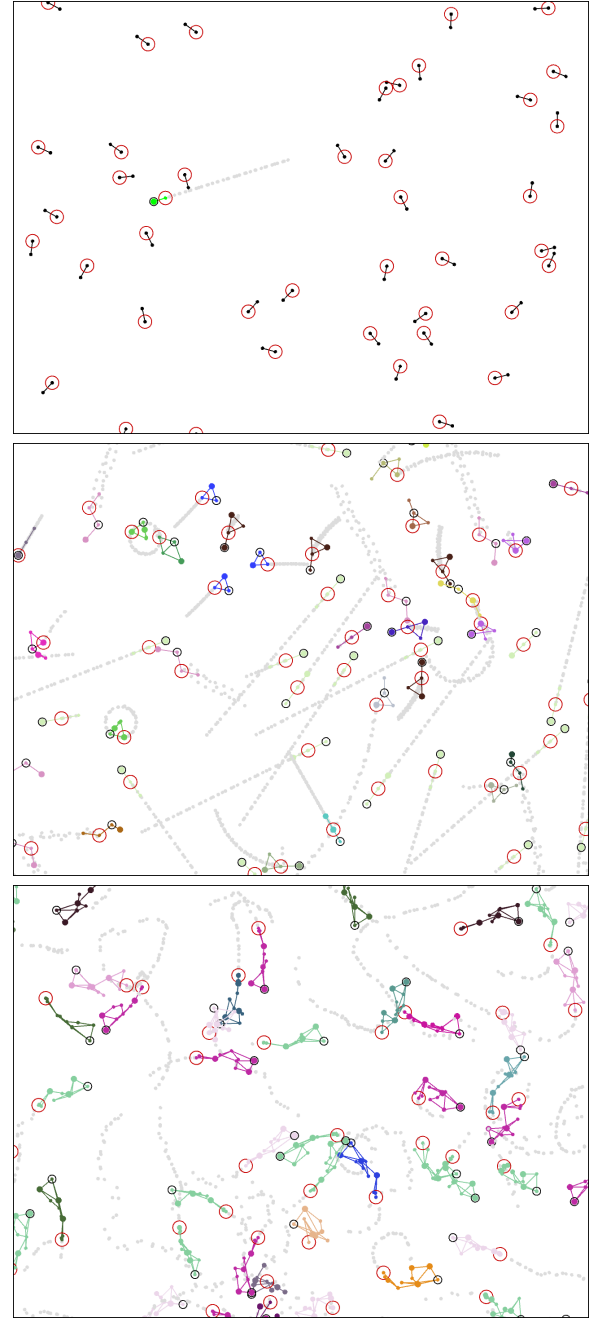

Figure 3: *The initial state of the simulator, with a single green moving creature (top), and later snapshots of such a simulation run (middle, bottom).*

At the start of the simulation, the lone moving creature crawls through an environment that is filled with stationary creatures, as in Figure 3 (top). This moving creature will be the ancestor of all the subsequent creatures in the simulation. At some point, this ancestral creature strikes the heart of a static creature. The static creature dies, and the ancestral creature is replicated. Further encounters with static creatures occur, and more single-segment creatures are born. All of these early creatures only travel in a straight line. The 2D environment is a rectangle with toroidal boundary conditions, so that a creature that wanders off one side of the world re-appears on the opposite side.

At some point, one of the early creatures is replicated with mutation, and then there is variation in the creature population. As the proportion of moving creatures increases, encounters between pairs of moving creatures start to occur. Encounters between identical one-segment creatures are won based on their relative positions and orientations (that is, essentially at random). Encounters between creatures with different body plans are more interesting, since there is the possibility that one of the creatures is more likely to win based on its body plan and behavior.

The creatures that evolve are different each time a lone ancestor simulation is run, due to using different random number seeds. (A typical simulation to 2,000,000 time steps requires roughly 4 hours of simulation time on a single 2.8 GHz processor.) Some general trends in creature success become apparent by observing the creatures in such runs. First, it is an advantage for a creature to move fast. Faster motion implies more frequent encounters with other creatures, and thus more opportunities to reproduce by winning such encounters. A pair of commonly successful features is to have the mouth near the front of the creature and have the heart near the back with respect to the direction of motion. Having a forward-positioned mouth means that this creature will be more likely to strike another creature's heart first, before that other creature has an opportunity to do so. A similar reasoning holds for the advantage of having a rear-positioned heart. Related to this is that many successful creatures cause their mouth to wave back and forth rapidly. This is an advantage because such a moving mouth is more likely to strike another creature's heart. Conversely, the motion of the heart in a successful creature is typically quite damped in comparison, and the heart is often dragged by a segment that has little or no oscillatory motion.

There is remarkable variation in body plans for fast moving creatures. In order to get a sense of the variation between runs, we performed 100 such lone ancestor simulations. Figure 4 shows the bodies of the most successful creatures from these 100 lone ancestor runs. Some creatures have elongated worm-like bodies, and they coordinate their segment oscillations to make rapid forward progress. Other creatures are composed of one or more triangles, and often such creatures seem to pulse in a manner that helps their forward progress
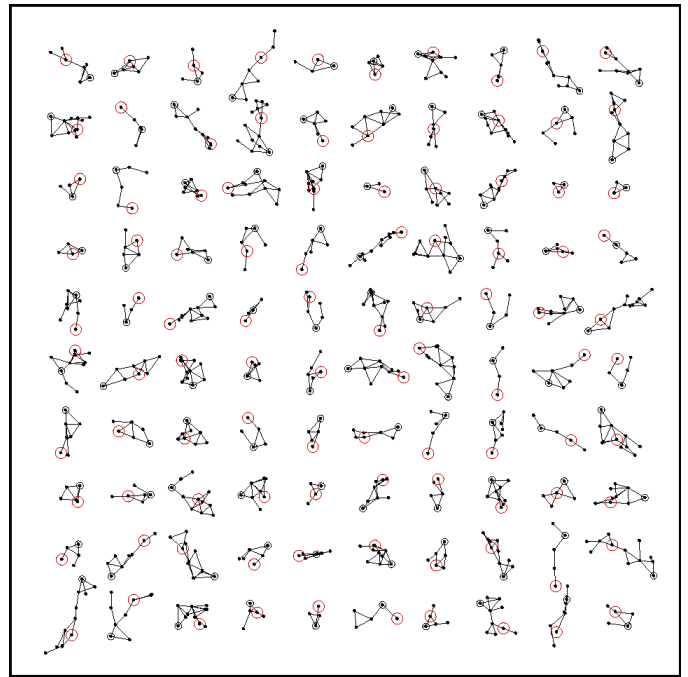

Figure 4: *The most successful creatures from 100 different lone ancestor simulations. Each represents the most numerous type of creature at time step 2,000,000 for a particular simulation.*

while at the same time causing their mouth to swing back and forth. Some creatures do not move in a straight line, but instead rotate in a circle, usually quite rapidly. Some creatures have triangles that form a compact body, but also have one or more "legs" that help to push them forward. Some of these legged creatures move with a limping gate, while others move in a smoother manner. One effective mode of locomotion is to have two trailing segments whose oscillation periods are offset from each other, so that while one segment is shortening and pushing the body forward, the other segment is elongating in the recovery phase of its duty cycle. The trails of creatures in Figure 3 (middle) illustrates some of the variations in motions of different creatures.

There is a limit to how fast a creature can move, given that there is a limit on segment lengths and on the frequency and magnitude of segment oscillations. There is, however, another avenue for creature evolution, and that is the ability to sense and react to other creatures. In many lone ancestor simulations, eventually some creatures arise that will turn their bodies towards the heart of another creature. This is the beginning of hunting behavior. Early in the development of this trait, a creature typically can only sense and turn to one side (e.g. just towards the right). Even more successful creatures are ones that can sense and turn towards creatures on either side. There is considerable room for fine-tuning this hunting behavior, including adjusting the placement and radius of the sensors, and modifying the magnitude of the
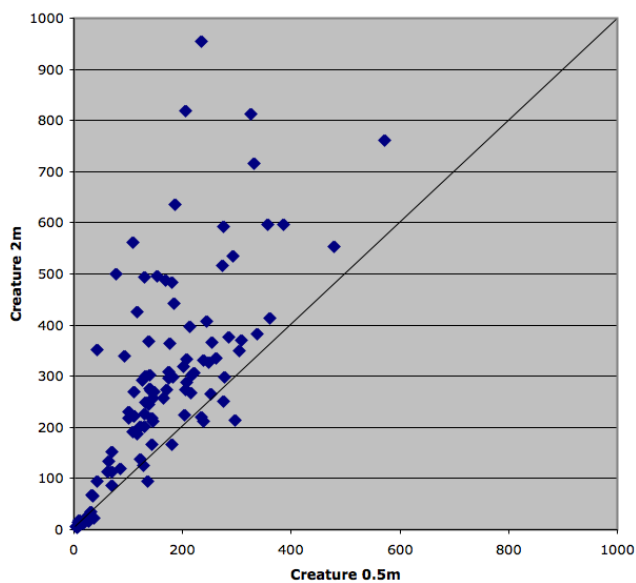
Figure 5: *Results from 100 between-generation creature contests. The horizontal axis is the number of captures by the earlier generation creature (time step 0.5 million), and the vertical axis is the capture count for the later generation creature from the same simulation run (time step 2 million).*

turning response when a sensor is triggered. Figure 3 (bottom) shows creatures that have evolved hunting behaviors.

Contests Between Generations

Although it appears to a human observer that later generations of creatures are more successful than earlier creatures, we used between-creature contests to determine whether this is in fact the case. Specifically, each of these contests is between two creatures that both evolved in the same simulation run. In a given contest, one of the creatures is the most numerous from time step 500,000 and the other is the most abundant creature from the same simulation run at the later time of 2,000,000. The goal is to see which of the two creatures can score the most captures in a fixed number of time steps. We ran 100 of these contests, one for each lone ancestor simulation run.

At the start of each between-generation contest, there are 50 copies of each creature. The rules of reproduction are modified so that this 50-to-50 ratio is always maintained throughout the contest. Instead of reproducing the victor of a creature encounter, the loser is removed from its current location and placed at a random location elsewhere in the environment. This transportation of the loser is performed regardless of whether a creature captures a different kind of creature, or whether it captures a replica of itself. There is also no mutation during the contest. A count is kept of the number of captures for each of the two types of creatures, and this count ignores same-type captures.

Figure 5 reports on the results of the 100 between-generation contests. Each point represents one contest. The horizontal axis is the number of creature captures by the 0.5 million time step creature (the earlier creature) and the vertical axis is the number of captures by the 2 million time step creature. Points below the diagonal line indicate more captures by the earlier creature, and points above the diagonal indicate that the later creature had more captures. There were 11 contests that were won by the earlier creature, 87 won by the later creature, and 2 ties. Note that later generations often made substantially more captures in many of the contests. We take this as verification that our rules for capture and reproduction are indeed effective at evolving creatures that are better suited for survival in a multi-creature environment.

Tournament Across Simulations

Although all of the creatures that evolved from the lone ancestor runs appeared to have adaptations for survival, there was a considerable variation in their modes of locomotion and their behavior. We wanted to find how these creatures from different simulations compared to each other when placed in the same environment. In order to explore this, we created a two-tier creature tournament. The first tier consisted of 10 contests, with 10 creatures in each contest. The 10 winners from these contests advanced to the second tier, where these creatures competed in a final contest that had a single victor. Figure 6 shows a frame from such a second tier tournament.

Each of the contests in the tournament began with 10 different types of creatures, and 10 copies of each of these creature types. The contest rules in this tournament differ from the between-generation contests. In particular, the winner of each encounter is copied, causing some creature types to become less or more numerous over time. No mutations occur during these contests. A contest ends when one type of creature is the lone survivor.

The bright green creature in Figure 6 is the tournament winner. As judged by these tournament, this is the most effective predator from the 100 lone ancestor simulation runs. This creature has 10 mass points and 13 segments. The body of this creature exhibits several innovations that evolved in order to make it a success. These innovations include a mouth that swings from side-to-side, a heart that is positioned on a "tail" that is dragged behind, the overall coordination between the oscillating segments that propels it forward, and sensors on both sides that steer it towards prey. The creature had been molded into this form by its numerous encounters with other creatures. In its own simulation environment, this creature is more deadly than all of its rivals. Nevertheless, the most simple real-world bacteria cell is still vastly more complex than this artificial creature. Despite this wide gulf in complexity, we believe that our results give an indication that multi-creature physical simulations can bring Artificial Life closer to simulating open ended complexity.
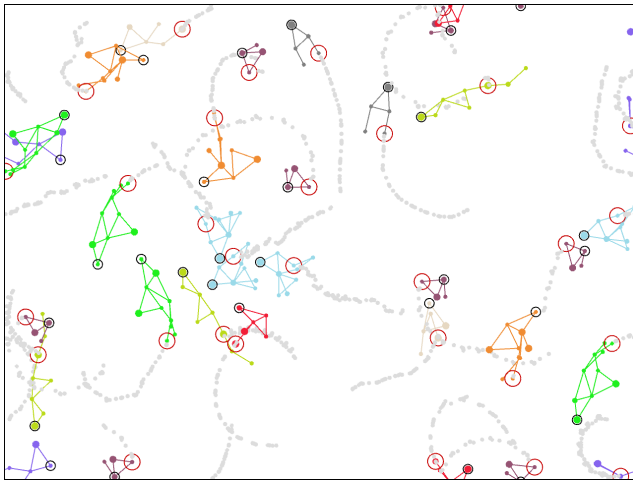
Figure 6: *Tournament between the best 10 creatures. For better visibility, only three of each creature has been placed into the environment. The bright green creature is the tournament winner.*

Conclusion and Future Work

We make two claims of novelty in our approach to simulated evolution:

- We simulate locomotion by dynamically changing the friction at either end of an oscillating spring.

- Our simulator combines pursuit/evasion behavior with the ability to evolve new physical configurations for locomotion.

A third important attribute of our simulator, shared by other researchers (Ventrella, 1996; Miconi, 2008), is that our simulated life-forms evolve in a large multi-creature environment that is driven by a simple physics engine. Taken together, these attributes create a rich synthetic environment for the evolution of artificial creatures.

There are several logical avenues for future research. First, there are other physical attributes that the virtual creatures could use to broaden their styles of locomotion even further. Oscillating torsional springs is one such possible addition. Another direction would be to add a more realistic energy model to the simulator. Still another fruitful avenue would be to replace the asexual reproduction model with sexual reproduction. Finally, it would be interesting to add a developmental process to our creatures, since some researchers have found that more successful body plans can result (Komosinski and Rotaru-Varga, 2001).

Acknowledgements

References

Baraff, D., Kass, M., and Witkin, A. (1999). Physically Based Modeling, Course Notes. *Course 36, SIGGRAPH 1999*.

Braitenberg, V. (1984). *Vehicles: Experiments in synthetic psychology*. MIT press Cambridge, MA.

Burton, E. (2007). Sodaplay constructor. *http://www.sodaplay.com/*.

Komosinski, M. and Rotaru-Varga, A. (2001). Comparison of different genotype encodings for simulated three-dimensional agents. *Artificial Life*, 7(4):395–418.

Lipson, H. and Pollack, J. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978.

Miconi, T. (2008). Evosphere: evolutionary dynamics in a population of fighting virtual creatures. In *IEEE Congress on Evolutionary Computation 2008*, pages 3066–3073.

Miller, G. (1988). The motion dynamics of snakes and worms. In *SIGGRAPH '88*, pages 169–173.

Miller, G. and Cliff, D. (1994). Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In *From animals to animats 3*.

Ngo, J. and Marks, J. (1993). Spacetime constraints revisited. In *SIGGRAPH '93*, pages 343–350.

Reynolds, C. (1994). Competition, coevolution and the game of tag. In *Proc. Artificial Life 4*, pages 59–69.

Sims, K. (1994a). Evolving 3D morphology and behavior by competition. In *Proc. Artificial Life 4*, pages 28–39.

Sims, K. (1994b). Evolving virtual creatures. In *SIGGRAPH '94*, pages 15–22.

Taylor, T. and Massey, C. (2001). Recent developments in the evolution of morphologies and controllers for physically simulated creatures. *Artificial Life*, 7(1):77–87.

Van de Panne, M. and Fiume, E. (1993). Sensor-actuator networks. In *SIGGRAPH '93*, pages 335–342.

Ventrella, J. (1996). Sexual swimmers: Emergent morphology and locomotion without a fitness function. *From animals to animats*, 4:484–493.

Yaeger, L. (1994). Poly world: Life in a new context. In *Proc. Artificial Life III*, pages 263–263.