

Evolutionary Dynamics Discovered via Visualization in the BREVE Simulation Environment

Lee Spector

Cognitive Science
Hampshire College
Amherst, MA 01002, USA
lspector@hampshire.edu

Jon Klein

Physical Resource Theory, Chalmers U. of Technology
and Göteborg University, SE-412 96 Göteborg, Sweden
also: CS, Hampshire College, Amherst, MA 01002, USA
jklein@hampshire.edu

Abstract

We report how BREVE, a simulation environment with rich 3D graphics, was used to discover significant patterns in the dynamics of a system that evolves controllers for swarms of goal-directed agents. These patterns were *discovered* via visualization in the sense that we had not considered their relevance or thought to look for them initially, but they became obvious upon visually observing the behavior of the system. In this paper we briefly describe BREVE and the system of evolving swarms that we implemented within it. We then describe two discovered properties of the evolutionary dynamics of the system: transitions to/from genetic drift regimes and the emergence of collective or multicellular organization. We comment more generally on the utility of 3D visualization for the discovery of biologically significant phenomena and briefly describe our ongoing work in this area. Pointers are provided to on-line resources including source code and animations that demonstrate several of the described effects. Associated links are available on-line at <http://hampshire.edu/lspector/alife8-visualization.html>.

Visualization and Scientific Discovery

Scientific discoveries often stem from unexpected observations: a scientist measuring one aspect of a system stumbles across an unusual feature of another aspect of the system, leading to further investigation and the discovery of previously unimagined phenomena. For this reason it is important for scientists to observe their experiments as broadly as possible; one cannot discover the unexpected if one's observational focus is too narrow.

On the other hand, many experimental frameworks, including many artificial life simulations, generate huge volumes of data that can easily overwhelm any observer. The challenge is to find ways of presenting this data that are comprehensible and yet sufficiently broad to allow for discovery of the unexpected.

For many artificial life simulations a simple but nonetheless powerful approach can be to depict the simulated world literally in high-resolution, real-time 3D animation. Human visual systems are well adapted to the

recognition of biologically significant behaviors patterns in dynamic, 3D views of the world, and we can therefore expect literal 3D animation to be useful for discovering such patterns in simulations within which biologically relevant features have visual representations.

In this paper we describe our experience using literal 3D visualization to discover unexpected features of an artificial life simulation. In the next two sections we describe our simulation and visualization system (BREVE) and the particular simulation from which our later examples are drawn. We then turn to two unexpected discoveries concerning the evolutionary dynamics of the system that were made possible by the visualization. Before we conclude we detail ongoing work that extends the simulation described here, with which we hope to make additional discoveries.

BREVE

BREVE is a simulation package designed for realistic simulations of decentralized systems and artificial life in 3D worlds. Simulations are written by defining the behaviors and interactions of agents using a simple object-oriented programming language called "steve." BREVE provides facilities for rigid body simulation, collision detection/response, and articulated body simulation. It simplifies the rapid construction of complex multi-agent simulations.

Although BREVE was originally developed to simulate "realistic" 3D worlds, it has proven to be useful as a visualization tool for abstract data and evolutionary computation algorithms as well. BREVE includes a powerful OpenGL display engine that allows observers to manipulate the perspective in the 3D world and view the agents from any location and angle.

In addition to the visualization benefits inherent in viewing data in a 3D world, the display engine also provides several "special effects" which can provide additional visual cues to observers. These effects include shadows, reflections, lighting, semi-transparent bitmaps, lines connecting neighboring objects, texturing of objects

and the ability to treat objects as light sources. All of the parameters and effects associated with visualization can be programmatically modified and may change dynamically over the course of a simulation.

More information about BREVE can be found in (Klein 2002). The BREVE system itself can be found on-line at <http://www.spiderland.org/breve>.

SWARM and SWARMEVOLVE in BREVE

One of the demonstration programs distributed with BREVE is SWARM, a simulation of flocking behavior modeled on the “boids” work of Craig W. Reynolds (Reynolds 1987). In the SWARM program the acceleration vector for each agent is determined at each time step via the following formulae:

$$\mathbf{V} = c_1 \mathbf{V}_1 + c_2 \mathbf{V}_2 + c_3 \mathbf{V}_3 + c_4 \mathbf{V}_4 + c_5 \mathbf{V}_5$$

$$\mathbf{A} = m \left(\frac{\mathbf{V}}{|\mathbf{V}|} \right)$$

In the first formula all of the c_i are constants and all of the \mathbf{V}_i are vectors determined from the state of the world (or in one case from the random number generator) and then normalized to length 1. \mathbf{V}_1 is a vector away from neighbors that are within a “crowding” radius, \mathbf{V}_2 is a vector toward the center of the world, \mathbf{V}_3 is the average of the agent’s neighbors’ velocity vectors, \mathbf{V}_4 is a vector toward the center of gravity of all agents, and \mathbf{V}_5 is a random vector. In the second formula we normalize the resulting velocity vector to length 1 (assuming its length is not zero) and set the agent’s acceleration to the product of this result and m , a constant that determines the agent’s maximum acceleration.

By setting different values for the c_i and m constants (along with the “crowding” distance, the number of agents, and other parameters) one can obtain a range of different flocking behaviors; many researchers have explored the space of these behaviors since Reynolds’s pioneering work.¹

In order to explore the evolution of controllers for heterogeneous swarms of goal-directed agents we made a number of enhancements to the SWARM program in BREVE, producing a program that we call SWARMEVOLVE. The first enhancement was to create several (typically three) distinct species of agents, each designated by a different color.² As part of this enhance-

¹The BREVE SWARM program also models a floor and hard-coded “land” and “take off” behaviors, but these are peripheral to the focus of this paper. Note also that neither this simulation nor the enhancement described below take full advantage of BREVE’s physical simulation engine — velocities are determined directly from the motion formula without regard to physical collision response, gravity, friction, etc.

²The species distinctions here are hard-coded, although in current work we are allowing for the dynamic emergence of species-like groups; see the “Ongoing Work” section.

ment we added a new term, $c_6 \mathbf{V}_6$, to the motion formula, where \mathbf{V}_6 is a vector away from neighbors of *other species* that are within a “crowding” radius.

Goal-orientation was introduced by adding a number (typically two) of randomly moving “energy” sources to the environment and imposing an energy dynamics. As part of this enhancement we added one more new term, $c_7 \mathbf{V}_7$, to the motion formula, where \mathbf{V}_7 is a vector toward the nearest energy source. Each time an agent collides with an energy source it receives an energy boost (up to a maximum), while each of the following bears an energy cost:

- Survival for a simulation time step (a small “cost of living”).
- Collision with another agent.
- Being in a neighborhood (bounded by a pre-set radius) in which representatives of the agent’s species are outnumbered by representatives of other species.
- Giving birth (see below).

The numerical values for the energy costs (along with those for “crowding” radii and other parameters) can be adjusted arbitrarily; values typical of those that we used can be found in the source code for our program, which is available from <http://hampshire.edu/lspector/swarめvolve-1.0.tz>.

As a final enhancement we leveraged the energy dynamics to provide a “fitness” function and used a genetic encoding of the control constants to allow for evolution. Each individual has its own set of c_i constants³; this set of constants controls the agent’s behavior (via the enhanced motion formula) and also serves as the agent’s genotype. When an agent’s energy falls to zero the agent “dies” and is “reborn” (in the same location) by receiving a new genotype and an infusion of energy. The genotype is taken, with possible mutation (small perturbation of each constant⁴) from the “best” current individual of the agent’s species (which may be at a distant location).⁵

³The constants are normally constrained to a minimum of 0.1 and a maximum of 15.0

⁴Perturbation values are typically between -0.5 and 0.5 (uniformly distributed).

⁵We chose to have births occur at the location of the prior death both for simplicity and because our first attempts to have births occur near parents produced jerky animation; the automatic camera control routine computes the camera parameters from the average agent position, and this changes suddenly if an agent dies in one place and another is born elsewhere. The choice to have death and rebirth happen in the same location facilitated, as an unanticipated side effect, the evolution of the simple form of multicellularity described below. In our ongoing work (also described below), however, we have moved closer to biology by having births occur near

We define “best” here as the *product* of energy and age (in simulation time steps). The genotype of the “dead” agent is lost, and the agent that provided the genotype for the new agent pays a small energy penalty for giving birth.⁶

The visualization system simply presents a 3D view (automatically scaled and targeted) of the geometry of the world and all of the agents in real time. Commonly available hardware is sufficient for fluid action and animation. Each agent is a cone with a pentagonal base and a hue determined by the agent’s species (red, blue, or purple). The color of an agent is dimmed in inverse proportion to its energy — agents with nearly maximal energy glow brightly while those with nearly zero energy are almost black. “Rebirth” events are clearly visible as agents flash from black to bright colors.⁷

Agent cones are oriented in the direction of their velocity vectors. This often produces an appearance akin to swimming or to “swooping” birds, particularly when agents are moving quickly.

Energy sources are flat, bright yellow pentagonal disks that hover at a fixed distance above the floor and occasionally glide to new, random positions within a fixed distance from the center of the world. An automatic camera control algorithm adjusts camera zoom and targeting continuously in an attempt to keep most of the action in view.

Figure 1 shows a snapshot of a typical view of the SWARMEVOLVE world. An animation showing a typical action sequence can be found on-line at <http://hampshire.edu/l spectator/swarmevolve-ex1.mov>.

The SWARMEVOLVE program that results from these enhancements is still simple in many respects but it nonetheless exhibits rich evolutionary behavior. For example, each species serves as an independent breeding population but the species interact via contention for space at the energy sources. One can often observe the species adopting different strategies; for example, one species often evolves to be better at tracking quickly moving energy sources, while another evolves to be better at capturing static energy sources from other species. An animation demonstrating evolved strategies such as these can be found on-line at <http://hampshire.edu/l spectator/swarmevolve-ex2.mov>.

parents and we have throttled the automatic camera control routine to ensure smooth animation.

⁶Reproduction is asexual in the system described here, but not in the ongoing work briefly described at the end of this paper.

⁷Birth energies are typically chosen to be random numbers in the vicinity of half of the maximum.

Genetic Drift Regimes

Many evolutionary systems are subject to genetic drift: random genotypic changes that are not subject to selection persist and populations diverge in the areas of their genotypes in which such changes occur (Wright 1997; Kimura 1997). While it would make sense to look for and to measure genetic drift in almost any evolutionary computation system, we did not initially think to do so in SWARMEVOLVE. Upon initial observation of the system in action, however, it became clear that the geometry and dynamics of SWARMEVOLVE conspire to produce two distinct regimes with respect to drift.

When a species includes members who have recently fed at an energy source there is selection for behaviors that maximize such feeding. If, however, circumstances in the world (including movements of energy sources and other agents) prevent any member of a species from feeding over a sufficient period of time then selection becomes randomized. In these cases the entire species starts to behave in erratic ways, often clearly oblivious of the location of the energy sources or of the center of the world. Their genotypes drift randomly in genotype space and their bodies also sometimes, as a result, drift randomly in physical space.

The probability of entering this “drift regime” depends on system parameters such as the cost of living, the mutation rate, and the dynamism of the energy sources; we have not yet explored this parameter space systematically. In many of our simulations, however, we have observed species entering, and later leaving, the drift regime. A species will leave the drift regime only after an individual drifts to a successful feeding strategy and begins to bear children.

It is difficult to tell from a still image whether or not a species is drifting, but it is often obvious from visual observation of the system in motion. The “oblivious” and uncoordinated motion jumps out to the human eye.⁸ An animation demonstrating a species in the drift regime can be found on-line at <http://hampshire.edu/l spectator/swarmevolve-ex3.mov>; note the behavior of the blue agents. In some cases drift is visible at a glance because obliviousness to the center of the world allows agents to travel far from the center of activity, which in turn causes the automatic camera control mechanisms to zoom the camera far out. Figure 2 shows a snapshot of such a situation.

Emergence of Collectives and Multicellularity

Many SWARMEVOLVE runs produce at least some species that tend to form static clouds around energy

⁸Casual observers have reported, e.g., feeling sorry for “those poor, clueless blue ones.”



Figure 1: A view of SWARMEVOLVE 1.0 (which is in color but will print black and white in the proceedings). The agents in control of the pentagonal energy source are of the purple species, those in the distance in the upper center of the image are blue, and a few strays (including those on the left of the image) are red. All agents are the same size so relative size on screen indicates distance from the camera.

sources. In such a species a small number of individuals will typically hover within the energy source, feeding continuously, while all of the other individuals will hover in a spherical area surrounding the energy source, maintaining approximately equal distances between themselves and their neighbors. Figure 3 shows a snapshot of such a situation, as does the animation at <http://hampshire.edu/lsector/swarめvolve-ex2.mov>; note the behavior of the purple agents.

We initially found this behavior puzzling as the individuals that are not actually feeding quickly die. On first glance this does not appear to be adaptive behavior, and yet this behavior emerges frequently and appears to be relatively stable. Upon reflection, however, it was clear that we were actually observing the emergence of a higher level of organization.

When an agent dies it is reborn, in place, with a (possibly mutated) version of the genotype of the “best” current individual of the agent’s species, where quality is

determined from the product of age and energy. This means that the new children that replace the dying individuals on the periphery of the cloud will be near-clones of the feeding individuals within the energy source. Since the cloud generally serves to repel members of other species, the formation of a cloud is a good strategy for keeping control of the energy source. In addition, by remaining sufficiently spread out, the feeding species limits the possibility of collisions with itself (which have energy costs). The high level of genetic redundancy in the cloud is also adaptive insofar as it increases the chances that the genotype will survive after a disruption (which will occur, for example, when the energy source moves).

The entire feeding cloud can therefore be thought of as a genetically coupled collective, or even as a multicellular organism in which the peripheral agents act as defensive organs and the central agents act as digestive and reproductive organs.

Once again, it is possible that one might have antic-



Figure 2: A view of SWARMEVOLVE 1.0 in which genetic drift has lead to physical drift, which in turn has caused the automatic camera control mechanisms to zoom far out.

ipated this phenomenon prior to visualization, but we did not. Upon viewing the 3D animation of the world in action, however, the phenomenon was obvious.

Ongoing Work

The SWARMEVOLVE program is under active development and we expect discovery-by-visualization to continue to be an important part of our research program.

We are working to increase the complexity and realism of the simulated world. For example the energy sources in SWARMEVOLVE 1.5 are depleted by eating (shown in the visualization by fading to white), they re-grow their energy over time, and their signals (sensed by agents) depend on energy and decay over distance according to an inverse square law. Births now occur near mothers while dead agents leave corpses that fall to the ground and decompose. Figure 4 shows a snapshot of SWARMEVOLVE 1.5; an animation showing a typical action sequence can be found on-line at <http://hampshire.edu/l spectator/swarmevolve-ex4.mov>. Near term plans call for the addition of edible

corpses, emergent rather than hard-coded colors/species, and rich, flexible agent sensors. Facilities for changing important system parameters (for example mutation rate and energy source stability) and monitoring system metrics (for example food utilization) via the graphical user interface have also been added.

We are also working to allow for the evolution of arbitrary agent control programs, eliminating the use of the hard-coded swarm motion formula (for which, in the implementation described above, only the constants are subject to evolution). We are doing this by replacing the “string of constants” genotypes with executable programs expressed in the Push programming language, which was designed expressly for purposes such as this (Spector 2001; Spector & Robinson 2002). The use of Push here will allow for the evolution of agent control programs with arbitrary control and data structures, and will also allow agent reproductive mechanisms (which might, for example, evolve to make use of sexual recombination) to be encoded in their genotypes (as in the Pushpop system (Spector & Robinson 2002;



Figure 3: A view of SWARMEVOLVE 1.0 in which a cloud of agents (the blue species) is hovering around the energy source on the right. Only the central agents are feeding; the others are continually dying and being reborn. As described in the text this can be viewed as a form of emergent collective organization or multicellularity. In this image the agents controlling the energy source on the left are red and most of those between the energy sources and on the floor are purple.

Spector 2002)).

The cumulative result of these enhancements will be a program with vastly more complex behavior than that of the SWARMEVOLVE system described in the previous sections, and the need for good visualization will be correspondingly more acute. Because most of relevant complexity (with the exception of the genotypes) will be visually represented in the real-time, quasi-realistic 3D rendering, we expect that we will continue to make visualization-based discoveries about the dynamics of this system.

In related work we have used BREVE's sound production facilities to add auditory representations to events (such as feeding) in SWARMEVOLVE (Spector & Klein 2002). While this work is preliminary we expect it to provide additional insights about the evolutionary dynamics of SWARMEVOLVE and its successors. We have also begun to investigate the effects of allowing agents to

sense some of the actions of others, in effect giving them ears with which to sense the events that we are rendering in audio. We hope by means of this to explore questions about the evolution of auditory communication.

Conclusions

The work described in this paper demonstrates that real-time 3D visualization of artificial life systems can lead to significant discoveries about the behavior of those systems. This is a straightforward idea, more modest in conception than exotic visualization schemes that have sometimes been proposed, but literal 3D representation can be surprisingly helpful. We conjecture that the effectiveness of literal 3D visualization is due to the adaptation of the human visual system to the recognition of biologically significant patterns in dynamic, 3D visual representations. Currently available computer hardware and graphics programming tools (such as OpenGL) make

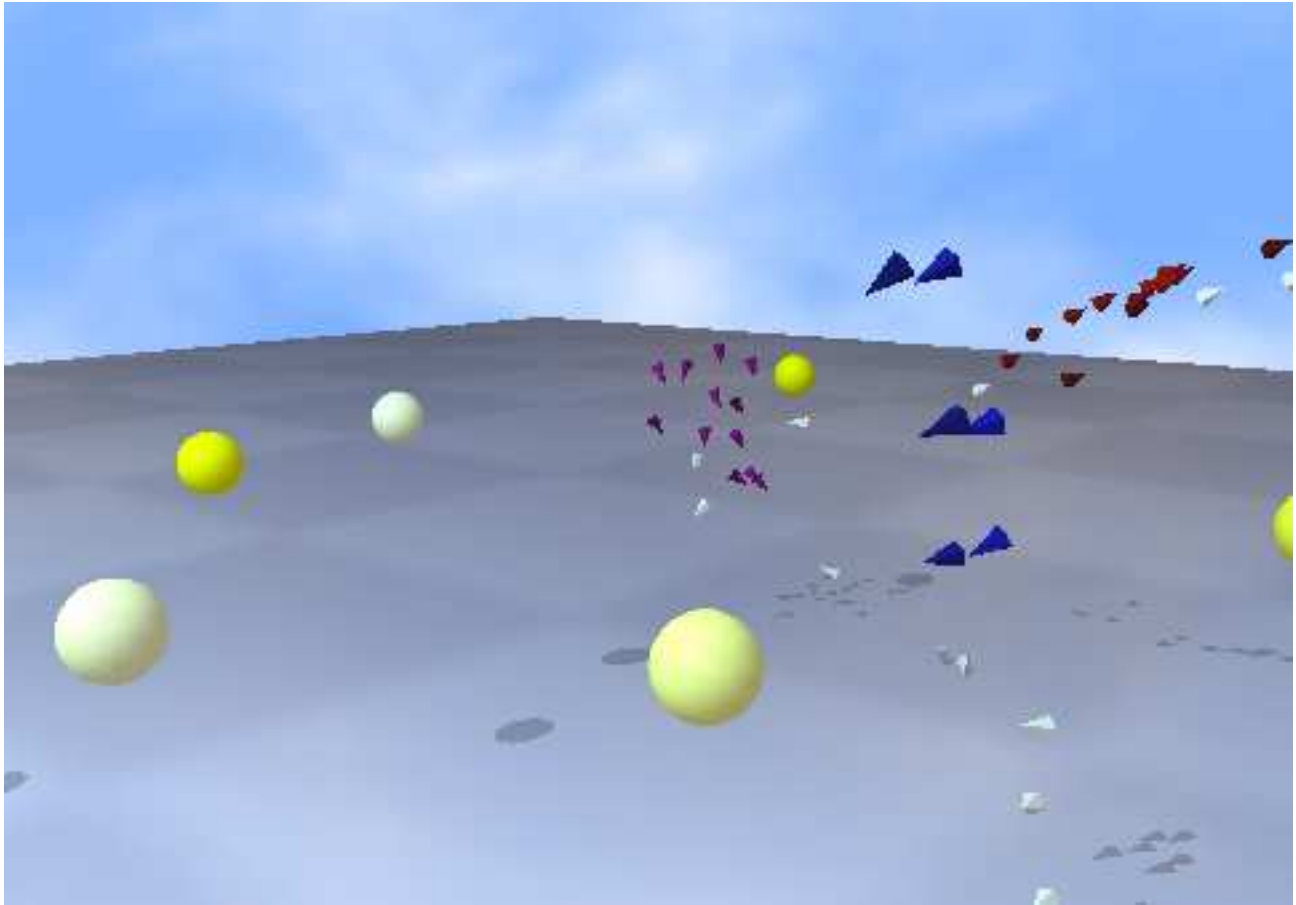


Figure 4: A view of SWARMEVOLVE 1.5, in which the color of the now-spherical energy sources indicates energy content; bright yellow sources (like the one second from the left) are full or nearly full while white sources (like the one furthest to the left) are empty or nearly empty. The white agents are corpses; these fall to the floor and disintegrate after a short period of time.

it possible to produce these kinds of visualization tools with relatively little effort; our recommendation is therefore that artificial life researchers consider adding such tools to their systems if they have not already done so.

Acknowledgments Thanks are due to Rebecca S. Neimark, Mark Feinstein, Alan Robinson, Ray Copinger, and Anton Fine.

This effort was sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30502-00-2-0611. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency

(DARPA), the Air Force Research Laboratory, or the U.S. Government. This research was also made possible by generous funding from Hampshire College to the Institute for Computational Intelligence at Hampshire College.

References

- Kimura, M. 1997. Recent development of the neutral theory viewed from the Wrightian tradition of theoretical population genetics. In Ridley, M., ed., *Evolution*. Oxford: Oxford University Press. 88–94. Reprinted from *Proc. of the National Academy of Sciences USA*, 88 (1991), 5969–73.
- Klein, J. 2002. BREVE: a 3d environment for the simulation of decentralized systems and artificial life. In *Proceedings of Artificial Life VIII, The 8th International Conference on the Simulation and Synthesis of Living Systems*. The MIT Press.
- Reynolds, C. W. 1987. Flocks, herds, and schools:

- A distributed behavioral model. *Computer Graphics* 21(4):25–34.
- Spector, L., and Klein, J. 2002. Complex adaptive music systems in the BREVE simulation environment. In Bilotta, E.; Lund, H. H.; Pagliarini, L.; and Pantano, P., eds., *Proceedings of the Workshop on “Artificial Life Models for Musical Applications II : Searching for musical creativity” at Artificial Life VIII, The 8th International Conference on the Simulation and Synthesis of Living Systems*.
- Spector, L., and Robinson, A. 2002. Genetic programming and autoconstructive evolution with the push programming language. *Genetic Programming and Evolvable Machines* 3(1):7–40.
- Spector, L. 2001. Autoconstructive evolution: Push, pushgp, and pushpop. In Spector, L.; Goodman, E.; Wu, A.; Langdon, W. B.; Voigt, H.-M.; Gen, M.; Sen, S.; Dorigo, M.; Pezeshk, S.; Garzon, M.; and Burke, E., eds., *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2001*, 137–146. Morgan Kaufmann Publishers.
- Spector, L. 2002. Adaptive populations of endogenously diversifying *pushpop* organisms are reliably diverse. In *Proceedings of Artificial Life VIII, The 8th International Conference on the Simulation and Synthesis of Living Systems*. The MIT Press.
- Wright, S. 1997. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In Ridley, M., ed., *Evolution*. Oxford: Oxford University Press. 32–40. Reprinted from *Proc. of the VI International Congress of Genetics*, I (1932), 356–66.