

Dynamic Habitat Construction Through Evolving 3D Swarms

Louise Magbunduku¹ and Kyle Harrington¹

¹Virtual Technology and Design, University of Idaho, Moscow, ID 83844
kharrington@uidaho.edu

Abstract

Exploration of inhospitable environments creates a need for the ability to remotely change the landscape so that it can support human life or necessary technology. The goal of this experiment involves re-purposing the mechanics and dynamics of an existing model of eco-evolutionary genetically-regulated swarms in order to construct a recognizable structure. In doing this we created a microscopic adaption of energy dynamics for swarm based models. This adaptation extends the energy reservoir system that is present in previous experiments. In this case the model uses one large energy source with microscopic alterable structure that yields energy for agents rather than multiple smaller energy sources dispersed throughout an environment. This allows us to study the microscopic dynamics of large scale agent interactions.

Introduction

The ability to trace agent pathways in swarm simulations presents countless possibilities in artificial life research. To this end, swarms have been shown to successfully execute dynamic growth processes such as L-systems and other architectural endeavors (Von Mammen and Jacob, 2009). This fashion of visualizing swarms dynamics has the possibility to engage a more utilitarian angle when approaching artificial life models. Instead of simply observing agent behaviors in a given ecology, the opportunity for more interactive and engaging modeling arises. To this end, we must take model flexibility into account. If agent behaviors prove to be too simple or the environment of the model proves to be too rigid it can lead to frustrating limitations in a project. Many modern swarm simulations use Craig Reynolds classic Boids model (Reynolds, 1987) as a starting point. Specifically, agents in most swarm models are designed to eventually evolve and exhibit the ability align with neighboring agent directions as well as the ability to find a common center of cohesion as well as repulsion from neighbors in extremely close proximity to avoid colliding. Additionally, there is interest in the use of swarms as a means of autonomous construction (Werfel, 2006). Previous work on swarm grammars have used swarm dynamics

to construct structures using a model drawing from swarms and L-systems (Von Mammen and Jacob, 2009).

To facilitate coherent construction there are a few features implemented into our model. These features bear resemblance to some of the approaches used in past swarm construction experiments (Werfel, 2006) for the intended use of avoiding common swarm construction pitfalls.

- There is an energy dense core initialized in the center of the simulation to initialize the population into a stable state.
- There is a spherical 3D framework initialized into the simulation which determines the bounds of habitat structural space in context of agent energy exchange with voxels.
- Agents receive more energy for depositing material into voxels that are closer to their target intensity. This is a way to replicate the natural inclination of termites to deposit materials where there has already been construction. (Werfel and Nagpal, 2006).

Using these rules as a framework for agent dynamics and evolution, we have been able to produce a number of coherent structures. (shown in Figure 1).

Termite Construction and Block Models

Termites mound building has been extensively studied in biological systems (Dangerfield et al., 1998; Korb, 2010). Termite construction is often referenced regarding swarm habitat building (Werfel, 2006). Termites are allogenic builders that, despite their small size, build towering structures to support their considerable population (Dangerfield et al., 1998). Given the size and complexity of these structures, this makes termite behavior an appealing choice for replication in a virtual setting. Block construction experiments have used this angle to build structures (Werfel and Nagpal, 2006). Our own model is very similar to a 3D block model. Voxels in the case would be analogous to blocks. Our model is different in the sense that voxel intensity is variable, which affects potential energy yield and thus agent decisions. Termites leave markers when they manipulate

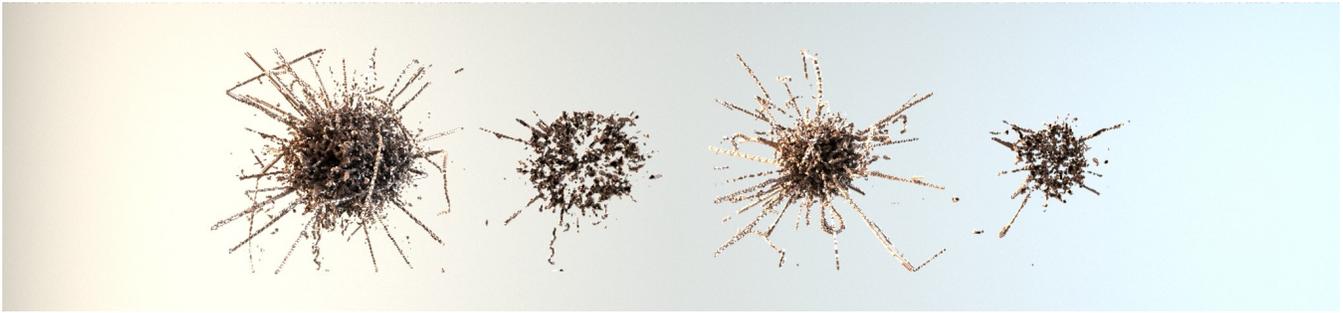


Figure 1: 3D render of model and cross section at 10,000th timestep.

a structure. These pheromones influence other termites to build upon their contributions. Our simulation emulates this behavior as agents are given the incentive to build in regions of activity. By making the energy yield of voxels determined by distance to target intensity, agents are inclined to be active in areas where other agents have been.

Model

In this work we extend the model previously presented in (Harrington and Magbunduku, 2017) to explore the microscopic dynamics of agent interactions. Our model uses Brevis for simulating kinetic swarms in 3D (Harrington and Stiles, 2017). Brevis is an artificial life simulator written in the functional programming language Clojure (Hickey, 2008). Physics timesteps are treated independently from agent behavior, where timesteps of the physics engine are managed by the ODE4J physics engine. Timesteps for agent behavior are set at 0.5 units of time.

The Brevis engine is capable of taking snapshots of where energy is being distributed in the model. These snapshots can be exported in tiff format and then be exported into a 3rd party image analysis software with a 3D viewer using the FunImageJ library (Harrington et al., 2017). The resulting 3D model from the compilation of tiff images can then be exported as an .stl and is viewable in any 3D software that supports that extension.

The Environment

The energy ecosystem of this model is based on a grid of 3D voxels. There are no designated energy reservoirs located in the model. Instead agents gain energy by interacting with voxels. Agents have the choice to deposit or harvest material from voxels within some specified bounds. If the agent performs the wrong action, then they receive no energy gain from the voxel they are interacting with. The model is initialized with a spherical 3D framework with a small target radius (5 spatial units), or a large target radius (10 spatial units). The agents can interact with any voxels within the bounds of the simulation (200x200x200 spatial units); however, they will receive no energy for interacting with any voxels outside of the simulation boundary. The simulation

is initialized with an energy rich core at the target intensity of 500. This is to balance out the energy dynamics of the simulation, and prevent a population explosion or extinction as the voxels around the area get closer to target intensity. In earlier simulations where there was no energy core the only initial action agents could perform in relation to the voxels in the environment was deposition, which proved problematic when trying to evolve digging behavior. Initially all voxels will yield some form of energy relative to their target intensity. A voxel with an intensity of 0 will yield the largest possible amount of energy when an agent deposits material. A voxel that has already met its target intensity will yield no energy for an attempt to deposit material. Once a voxel surpasses an intensity threshold, it begins to generate energy at which time it would be beneficial for an agent to harvest material from the voxel rather than deposit material.

Agents

Agents are represented in the model through 3D cones (2.2 spatial units in length with a base of 1.5 units). Their navigational and feeding patterns are controlled by a gene regulatory network. The gene regulatory network relies on a series of inputs regarding the agent's position and neighborhood. Agents have a limited neighborhood radius perception of 50. To evolve, agents must gain enough energy to pass on a combination of their genes and another parent's to their offspring.

Agents are placed in the simulation with an energy of 1. When initialized, they are placed somewhere on the simulation's target radius. They gain energy solely through interaction with the aforementioned voxels. Once an agent surpasses a breeding threshold of 1.1, it will produce a child with another agent in the simulation. The child is created in the vicinity of its mother. Selection for reproduction is based on age. Energy is split 50/50 between the parent agent and its child. Agents in the simulation are subject to a cost of living. They lose energy at a fixed rate of 0.0015 per time step. The two primary methods for an agent to receive energy are depositing material into a voxel, and digging or removing energy from a voxel. An agent has a base consumption rate of 0.25. Voxels within habitat bounds have a target intensity

of 500. A voxel will not yield energy for harvesting material until the deposition threshold (50) has been reached. When depositing material, agents add 10 to the target intensity. Once the target intensity has been reached, a voxel will no longer yield energy for deposition. Energy yield from depositing is habitat intensity multiplied by the consumption rate. This way, agents receive more energy from depositing material into voxels that are close to the target intensity. An agent receives energy from digging the same way, except rather than a target intensity, the agent will gain energy based on the voxels proximity to the core. Agents will receive more energy by removing voxels which are closer to the center of the simulation. Voxels that fall on the target radius will yield no energy from digging. Additionally, agents will not receive any energy from interacting with voxels outside the target radius.

The Gene Regulatory Network

The Gene Regulatory Network (GRN) is a system in place to control and evolve agent behavior. Much like a neural network, the GRN works off a series of inputs and outputs designed to influence how agents interact with their surrounding environment and neighbors. The most important change to the GRN in this model is that agents gain the ability to decide whether or not they want to deposit material or dig up existing material. The agents have the ability to see the energy yield from a voxel and will make their decision based on potential gain.

Results

Initialization and Habitat Development

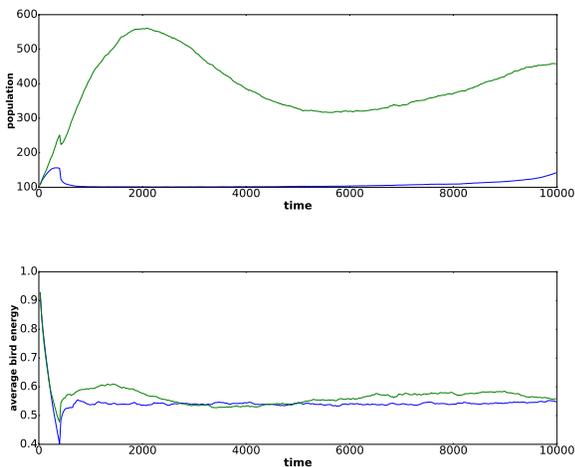


Figure 2: Population dynamics and average agent energy over time (Green = initial energy R[10], Blue = initial energy R[5]).

In Figure 2 we can see that at initialization, the population of agents climbs rapidly as the swarm interacts with the energy dense core. The population of both simulations then takes a sharp dip as the potential energy yield from the core decreases. For the larger environment, the population dip doesn't occur until around 2,000 timesteps. The population never dips below 300. In the smaller environment, this occurs much more quickly. The population spikes and falls to minimum before 1,000 timesteps have even passed. After about 4,000 timesteps, agents in the smaller environment adapt and can recover their population, and continually maintain above the minimum number of agents.

Agents begin the simulation by constructing a spherical structure in the center of the simulations axis, where the initial core is located. They begin by depositing and harvesting material from the voxels in the core, until the energy yield becomes too low to sustain a population (shown in Figure 3). From there, agents expand from the center outwards to the target radius. While expanding they maintain a spherical structure with a more dense exterior, and a lower intensity interior. From Figure 3 it is evident that there is a greater amount of energy being gained from deposition. However, in both the large and small environment the trend of energy harvest mirrors energy deposition nearly perfectly which explains why the habitats seem to maintain their general topology as their area expands.

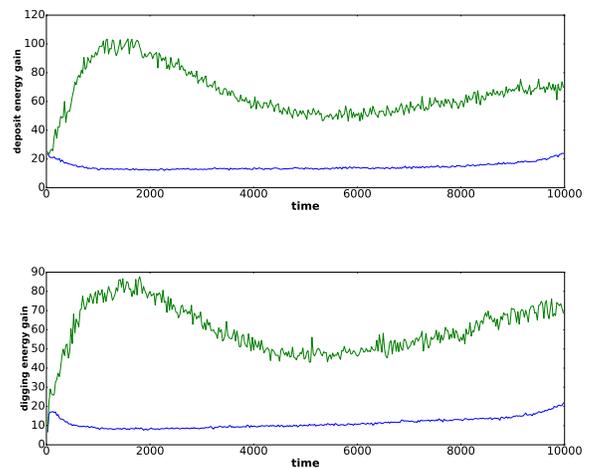


Figure 3: Temporal dynamics of energy gain from deposition and digging of material (Green = initial energy R[10], Blue = initial energy R[5]).

Mound Building and Tunneling

Although the agents in this model fly by nature, their method of construction tends to be very localized. The majority of agents start the construction process from the center and expand the habitat outward. Even though they are able to

fly their method of building is more akin to termite style mound building (Dangerfield et al., 1998). Since agents are rewarded by depositing materials in voxels with high neighboring intensity, the method in which they build is very structured (shown in Figure 4). Most agents will not attempt to skip ahead and build outside of the main central mass. When inspecting the 3D models, there is evidence of tunneling or pocketing behavior.

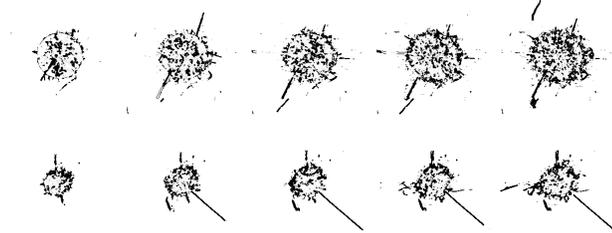


Figure 4: 3D render of cross section density changes at 2k, 4k, 6k, 8k, and 10k timesteps.

Agent Topology

Agents show a strong level of cohesion when the simulation initializes as they all move from the outside of the simulation's target radius towards the core (shown in Figure 5). Once the energy initial energy of the core is depleted, agents once again begin to expand back outward. Since agents have very similar patterns in regards to depositing and digging material, this might indicate that a larger number of agents are constructing the shell, while other agents are digging up what they are leaving behind. It's notable that the centroid average stays mostly stable throughout the simulation's run. This reinforces the observation that agents are growing and changing the habitat at the same rate. With a target radius of 10 there appears to be a gradual increase in cohesion among agents which may indicate that as the potential energy yield in the simulation for depositing, more agents are evolving to dig. Note that in Figure 3 the total amount of energy birds receive from digging and depositing material is nearly the same at 10k time steps. It is likely that more agents are moving back towards the center of the simulation as that becomes a more promising way to receive energy.

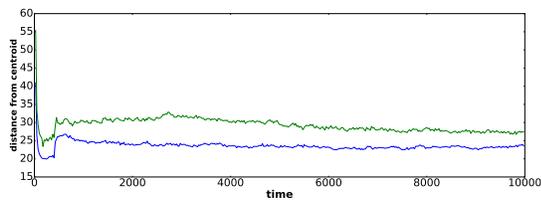


Figure 5: Average agent distance from centroid.

Conclusion

Here we present a model representation of how swarm intelligence can be used to construct structure, and possibly eventual livable habitats. The agents in this model are capable of building recognizable, defined structures. With the manner in which our model is set up the swarm agents will construct a habitat with an identifiable exterior and interior. Additionally, we have built a microcosmic environment which reflects the work done in previous models that incorporates one encompassing energy reservoir rather than several smaller resource point spaced throughout the simulation environment. Further exploration of these methods may demonstrate a way in which robotic swarm technology could autonomously construct a habitat in an environment inhospitable to humans.

Future Work

Collision

As previous experiments have shown that an increased collision cost leads to improved navigational and foraging abilities in swarms that have a genetically regulated network, it would be worthwhile to explore how that impacts feeding and energy gathering on a much smaller scale. Questions to address would be how well agents can survive within the space and how increased pressure through collision would impact the construction of habitats.

Conservation of Mass

To effectively present how this simulation could act as a proof of concept, and be translated from the virtual world, our next step would be introducing the law of Conservation of Mass. As it currently stands, agents from the simulation create materials to place in a voxel environment out of nowhere. To provide the same constraints physical technologies would find in a natural environment, we must supply for our agents an environment in which the only energy that can be manipulated and distributed is the energy initialized in the simulation. This could possibly be done by setting a "terrain" of voxels. From there agents can harvest material from this terrain and construct habitats from there.

Acknowledgements

This work was supported by the NSF Idaho EPSCoR Program and the National Science Foundation under award number IIA-1301792.

References

- Dangerfield, J. M., McCarthy, T. S., and Ellery, W. N. (1998). The Mound-Building Termite *Macrotermes michaelseni* as an Ecosystem Engineer. 14(4):507–520.
- Harrington, K. and Magbunduku, L. (2017). Competitive dynamics in eco-evolutionary genetically-regulated swarms. In *Proc. of the 14th European Conference on Artificial Life*, pages 190–197.

- Harrington, K. I., Rueden, C. T., and Eliceiri, K. W. (2017). Funimagej: a lisp framework for scientific image processing. page to appear.
- Harrington, K. I. S. and Stiles, T. (2017). kephale/brevis 0.10.4.
- Hickey, R. (2008). The clojure programming language. In *Proceedings of the 2008 symposium on Dynamic languages*, page 1. ACM.
- Korb, J. (2010). Termite mound architecture, from function to construction. In *Biology of Termites: A Modern Synthesis*, pages 349–373. Springer.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH computer graphics*, 21(4):25–34.
- Von Mammen, S. and Jacob, C. (2009). The evolution of swarm grammars-growing trees, crafting art, and bottom-up design. 4(3).
- Werfel, J. (2006). Anthills built to order: Automating construction with artificial swarms.
- Werfel, J. and Nagpal, R. (2006). Extended stigmergy in collective construction. *IEEE Intelligent Systems*, 21(2):20–28.