

Shape Restoration by Active Self-Assembly

Daniel Arbuckle and Aristides A. G. Requicha

*Laboratory for Molecular Robotics
University of Southern California
Los Angeles, California, USA*

{daniel.arbuckle | requicha} @ usc.edu

Abstract - Shape restoration is defined as the problem of constructing a desired, or goal, solid shape S_g by growing an initial solid S_i which is a subset of the goal but is otherwise unknown. This definition attempts to capture abstractly a situation that often arises in the physical world when a solid object loses its desired shape due to wear and tear, corrosion or other phenomena. For example, if the top of the femur becomes distorted, the hip joint no longer functions properly and may have to be replaced surgically. Growing it in place back to its original shape would be an attractive alternative to replacement. This paper presents a solution to the shape restoration problem by using autonomous assembly agents (robots) that self-assemble to fill the volume between S_g and S_i . If the robots have very small dimensions (micro or nano), the desired shape is approximated with high accuracy. The assembly agents initially execute a random walk. When two robots meet they may exchange a small number of messages. The robot behavior is controlled by a finite state machine with a small number of states. Communication on contact models chemical communication, which is likely to be the medium of choice for robots at the nanoscale, while small state and small messages are limitations that also are expected of nanorobots. Simulations presented here show that swarms of such robots organize themselves to achieve shape restoration by using distributed algorithms. This is one more example of an interesting geometric problem that can be solved by the active self-assembly paradigm introduced in previous papers by the authors.

Index Terms – self-assembly; swarm robotics.

I. INTRODUCTION

This paper is concerned with active assembly agents whose capabilities include a limited ability to exchange information with neighbors, to grip, release and move relative to neighbors, to perform a random walk through their environment, and to decrement a number. This small set of actions, controlled by a finite state machine, are sufficient to allow the solution of a large class of tasks through self-assembly. We have previously explored some of the capabilities of such assembly agents in [1].

In this paper, we address the problem of self-assembling a goal shape around a pre-existing object of unknown shape. We use Active Self-Assembly to achieve this goal.

The paper begins with a description of some related work, followed by a description of what we are trying to achieve and the general approach we have taken. Following

this are three sections describing our methods in more detail. Finally we discuss the results of applying these techniques in simulation, and our conclusions.

II. RELATED WORK

The concept of Active Self-Assembly was introduced by our USC colleagues in [2] and [3] and explored by the authors in [1].

Jones and Matarić [2] discussed a particular self-assembly technique that is applicable to any fully specified and fully connected structure in the plane. This technique places a state machine in each of the assembly agents. The assembly agents bind to one another when their states are compatible according to the transition rules of the state machine, and then transition to new states. The transition rules are automatically generated by a compiler which takes the desired planar structure as its input. This technique has at least one machine state for each agent position in the structure, and thus the memory requirement of the agents grows with the area of the structure.

Klavins, Ghrist and Lipsky [4] have described robotic self-assembly in terms of graph grammars and conformational switching, resulting in systems with capability and limitations comparable to those of Jones and Matarić.

Hormone-based control as described for example in [5] and [6] partially inspired the concept of Active Self-Assembly. The "hormones" presented in these papers are messages which trigger different actions in different places. Although these papers discuss self-reconfigurable robotics rather than self-assembly, the idea of explicit local communication is a powerful one for self-assembly swarms as well.

Wei-Min Shen, Peter Will and Berok Khoshnevis have applied hormone-based control to self-assembly, but in the context of spaceborne operation [3] where communication between non-neighbor assembly agents is practical, and complex assembly agents are acceptable. Neither assumption is maintained in this paper.

III. GOALS AND METHODS

Many structures must occasionally be rebuilt such that the resulting shape matches some specific desired shape. This is a hard task for self-assembly because the degradation of the old structure leaves an unpredictably shaped core in place. This core can interfere with the self-assembly process. Patterns of shape growth which expand outward from the core

The research reported in this paper was supported in part by the NSF under grants IIS-99-87977, EIA-01-21141 and DMI-02-09678, and Cooperative Agreement CCR-01-20778.

to create a predetermined shape seem to be generally unworkable. However, patterns which establish the boundary of the goal shape and then grow inwards toward the core can be made to work.

There are two stages to such a growth pattern: First, a seed must be located at a known position relative to the final desired shape. Second, the shape must be grown from the seed. This paper addresses only the second stage.

Establishing the boundary of a goal shape is a task addressable by Active Self-Assembly. This can be done by causing the assembly to take place along the path of a pre-routed message, as shown in the next section. Alternatively, if the environment provides cues about where the border should be, a partially specified shape [1] can be used to define the boundary.

Once the assembly agents create the boundary of the shape, the task becomes filling the interior. This is done by having the agents which make up the boundary shift themselves inward when they find themselves grabbing another agent on their 'out' side. The two agents move while bound together, and the outer agent becomes part of the shell while the inner agent becomes part of the interior of the shape. In this way, the shell acts as a one-way permeable membrane for assembly agents.

After performing the shift-inward operation, the agent which ends up on the inside can either release its hold and begin a random walk, or it can remain attached to the other agent. There are advantages and disadvantages to both choices.

The advantages of having the agents remain attached when they become part of the interior of the structure (the "guided method" below) are speed and information. Under this scheme, every agent which comes in contact with the structure during its construction can be incorporated into the structure within a constant amount of time. Further, it is possible to organize the construction so that the agents comprising the structure know when the construction is complete.

The advantages of having the agent release and wander when they become part of the interior of the structure (the "random method") are simplicity, completeness and repair. The state machine required to implement such a scheme is significantly less complicated than that of the guided method. Because the agents are free to wander anywhere where they are physically able to fit, given sufficient time the entire interior of the structure will be packed with assembly agents, whereas the guided method may leave gaps due to the constrained motion of the agents. Finally, the random method facilitates the creation of self-repairing structures.

IV. ESTABLISHING THE BOUNDARY

The boundary of the desired shape grows from an assembly agent called the "seed," which instigates the self-assembly process. If the environment permits it, we can use techniques such as the self-assembly of partially specified

shapes [1] to construct the boundary without resorting to encoding its precise shape in the self-assembly rules. If the environment is not so conveniently laid out, the assembly rules must fully specify the shape.

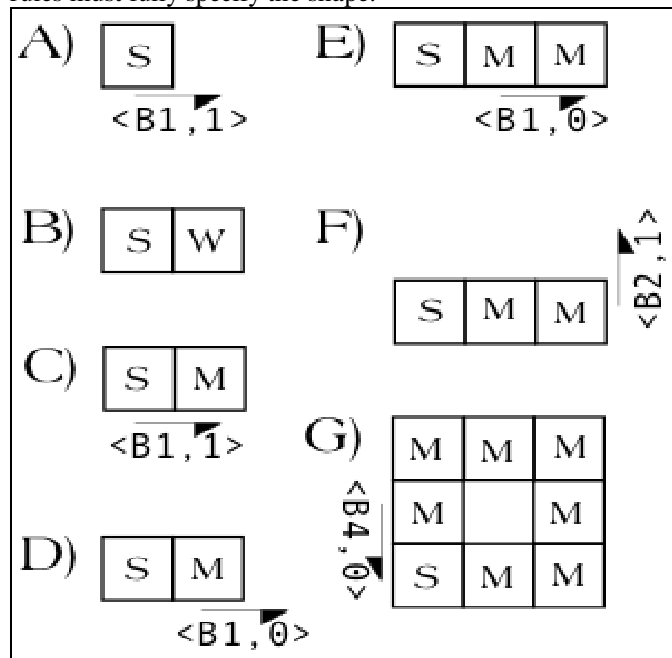


Figure 1

A good way for the assembly rules to specify a precise polygonal shape is to have them include a route along which the seed periodically sends a message. The message path defines the edges of the shape. Each time a message is to be passed to a neighbor which is not grabbed, this is a cue to grab that neighbor. Often there is no assembly agent in the neighboring position, in which case the message is dropped.

For example, consider assembling a 3x3 agent square boundary. Initially, the seed is alone, periodically attempting to send a message consisting of a tag B1 and a counter 1 (denoted <B1,1>) to a single neighbor (Fig. 1A).

After some time, another assembly agent's (agent *a1*) wandering through the environment causes it to become the neighbor to which the seed is transmitting (Fig. 1B).

This agent, on receiving <B1,1> changes state from "wander" to "member," i.e. from the state representing a random walk to the state representing membership in the structure (Fig. 1C).

When an agent in the "member" state receives a message with a counter greater than zero, the counter is decremented and the modified message is transmitted from the side opposite of where it was received (Fig. 1D).

Eventually another agent (agent *a2*) will become *a1*'s neighbor, receive the <B1,0> message *a1* is transmitting, and become a member (Fig. 1E).

Since <B1,0> has a counter of zero, the counter is not decremented and the message is not forwarded; instead, *a2* has a rule in its memory that says "when you are a member

and you receive <B1,0> from the left, remain a member and send <B2,1> up,” as shown in Fig. 1F.

Messages tagged B2 grow a vertical line and trigger the transmission of B3 messages, which grow a horizontal line and trigger the transmission of B4 messages. Once B4 messages have caused a line to grow into contact with the seed, the seed will start receiving <B4,0> messages, which indicates that the square is complete (Fig. 1G).

This way of defining shapes is different from the one presented in [1] in a couple of ways: this representation of a shape is not as compact, and so it requires more onboard memory in the assembly agents in order to implement, and the route-based representation can encode an approximation of any path, and so is more general. For example, the shells of the black constructions shown in Fig. 2 were grown in this way.

If the assembly agents have the ability to move relative to their neighbors, the propagation of the messages from agent to agent can provide a guide for the motion of assembly agents. This allows them to move toward the part of the structure which is currently being assembled. As the structure gains boundary area, more agents encounter it during their random walks through the environment and are guided to the unfinished part. This is a major efficiency gain, reducing the number of dropped messages and thus decreasing the time required to assemble the boundary.

This process generalizes to three-dimensional shapes. Again, the message routes define the edges and message propagation can guide assembly agents to where they are needed. Once the edges have been built, the guided method described in section VI can be used to fill in the faces of the shape, after which either the guided or random method can be used to fill in the interior of the shape.

To characterize the rate of growth of the border of a polygonal structure, we will assume that the assembly agents are cubes of unit dimension, that the volume within which the assembly is occurring is infinite, and that the density of assembly agents in the volume approaches a constant as distance away from the assembly structure increases to infinity. The angles between the sides of the polygon have a weak influence over the final result, and so for simplicity we will consider the polygon as a straight line with a length equal to the sum of the lengths of its sides.



Figure 2

The rate of growth is related to the rate at which assembly agents contact the growing structure. This rate is the sum of the rate of contact at the ends and the rate of contact in the rest of the line. The rest of this analysis assumes that the global population density of assembly agents remains approximately constant. At the ends of the line, the rate at which agents contact the structure is approximately constant, and we can treat it as such. In the middle, the contact rate at each member agent is approximately constant, and so the total contact rate grows with the number of assembly agents in the structure, up to the point where interference effects from the agents moving along the structure toward the construction area prevent further bindings. This interference occurs when the part of the structure closer to the seed is gathering assembly agents quickly enough that most messages are accompanied by an assembly agent.

Since new agents cannot take their place in the structure more often than messages reach the point of construction, and agents are reaching the point of construction at the same rate that messages are, and the rate of message transmission can be anything which makes sense for the hardware, we can see that the rate of growth increases to the maximum practicable for the assembly agent architecture as the structure grows.

Using routed messages to define polygonal or polyhedral shapes requires an amount of memory proportional to the number of vertices required to describe the shape, because each vertex requires a rule in memory describing in which direction to forward each message. A constant set of rules suffices to provide other necessary behavior.

V. THE RANDOM METHOD

The random method begins with a boundary (the “shell”) of the desired shape already constructed. The members of the shell are in a set of states we will collectively call “Shell.” One of the assembly agents (the “seed”) begins emitting messages (“seed messages”) to one of its neighbors. The state machines in the assembly agents contain a precomputed route for these messages, consisting of a sequence of “forward the message to neighbor X” instructions, which causes the message to visit each of the assembly agents in the shell and then return to the seed.

When an agent in the shell receives a seed message, and it is holding on to an assembly agent on its “out” side, it attempts to move inward while holding on to the outer agent, so that they both move in synchrony. If this succeeds, the outer agent will soon receive a seed message, which will trigger a transition to the shell state and cause it to release the agent on its “in” side, which will return to the initial random walk state. If the move inward fails, for example due to interference by another assembly agent already on the inside of the structure, the next receipt of a seed message will trigger another attempt.

If the seed messages used in the random method are the same messages which guide the formation of the shell, then

when the structure is damaged by removal of some of its non-seed members it will self-repair without any outside interference or guidance. For example, consider a square of sixteen assembly agents with the seed in the bottom left corner (Fig. 3A). Due to the action of a hostile environment, the bottom right four agents are knocked out of the structure (Fig. 3B). The seed messages trigger the same operations that constructed the shape originally, first placing assembly agents into the missing shell positions (Fig. 3C), then attempting to shift agents into the center (Fig. 3D).

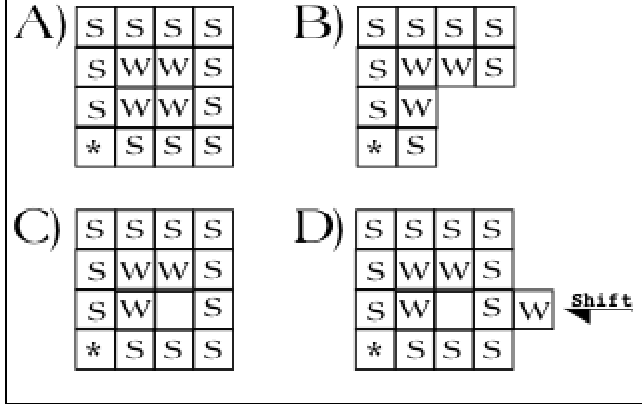


Figure 3

As the interior of the shape gets more crowded, the probability of interference in the shift inward operation increases. For this reason, the shape can not generally be guaranteed to fill entirely in any finite amount of time.

VI. THE GUIDED METHOD

The guided method, like the random method, begins with a shell of the desired shape already constructed.

When an agent in a Shell state receives a seed message, and it has hold of another assembly agent on its "out" side, it transitions to an Inner state and attempts to move inward while holding on to the outer agent. If this succeeds, the outer agent will soon receive a seed message, which triggers a transition to a Shell state. If the operation fails, the agent (now in an Inner state) will receive another seed message, telling it that there is no room for more agents on its "in" side. In that case, the agent transitions to a state from the set called Done. Regardless of whether the agent can move inward, a "shell message" is sent along the route instead of the seed message that was received.

When an agent in the Shell state receives a shell message, it pushes the agent grabbing its "out" side, if any, in the direction of propagation of the message, and send the message onward.

When an agent in the Done state receives a seed message, it pushes the agent grabbing it on the "out" side in the direction of propagation of the message, and sends the message onward.

Agents in corners change state from Shell to Done as soon as they receive a Seed message, and never try to shift inward.

The result is that when an assembly agent encounters the shell, it moves to the agent which is at the transition point between seed messages and shell messages, and is then incorporated into the structure. The transition point moves around the structure until it reaches the seed, at which time the seed is aware that the structure is complete, and can take whatever actions are appropriate for the continuing use of the structure. Example actions might be to trigger a transition to a new set of states in which the structure behaves differently, or to send a message which causes an assembly agent somewhere in the structure to become a seed for a new growth.

We can estimate the time required to fill in the structure using the guided method by examining the steady-state solution to Fick's equations (see for example [7]) describing the rate of diffusion to a spherical absorber in an infinite volume: $I = 4\pi DaC_0$ where I represents the rate of contact between assembly agents and the absorbing structure, D is the constant diffusion coefficient, a is the radius of the structure and C_0 is the concentration of assembly agents as the distance away from the structure approaches infinity. Since I and a are the only variables in this equation, the rate at which new assembly agents contact the structure is proportional to the radius, to the degree that a sphere approximates the shape of the structure. Because assembly agents make at most one complete traversal of the surface of the shape between their initial contact and the time that they are incorporated into the shape, which takes a constant amount of time per agent, the time required to fill the shape is approximately proportional to the volume of the shape over the rate at which assembly agents contact the surface, and is thus approximately proportional to the square of the radius of the shape: $\alpha_1 \frac{a^3}{I} = \alpha_2 a^2$.

VII. RESULTS

We have used Active Self-Assembly in simulated swarms to 'restore' irregular shapes into regular squares. Both the guided and random methods have been applied. The experiments were performed in a discrete-space 2D simulation with reflective boundaries at the edges of the "world." Each of the assembly agent was simulated individually, and simulations consisted of swarms of sizes ranging from 5000 to 20,000 or more assembly agents.

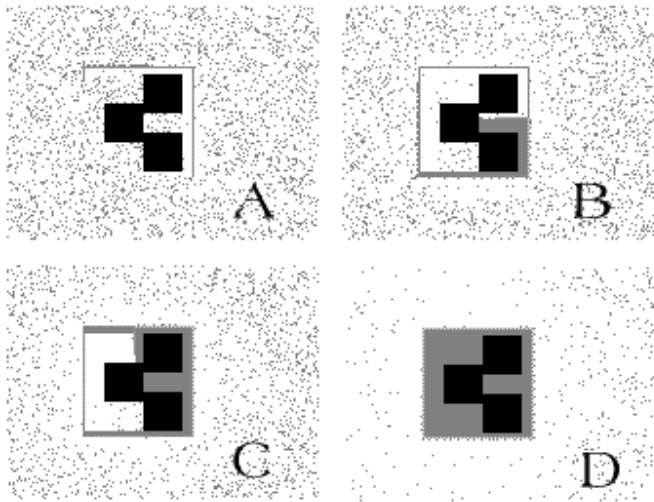


Figure 4

We found the guided method to be more difficult to implement and more susceptible to interference than the random method. However, it does live up to its promise of bounded assembly time, as can be seen in Fig. 5, a chart of the number of simulation steps required before the structure was completely filled over fifty consecutive assembly processes of the “R” shape in Fig. 2. Fig. 4 displays a guided self-assembly process in our simulator, (A) with the boundary partially complete, (B) in an early stage of assembly, with the transition from seed to shell messages partway up the right side, (C) in a later stage of assembly with the transition partway across the top, and (D) fully assembled. The black boxes in these figures represent the preexisting core around which we are assembling, while the light grey represents the assembly agents.

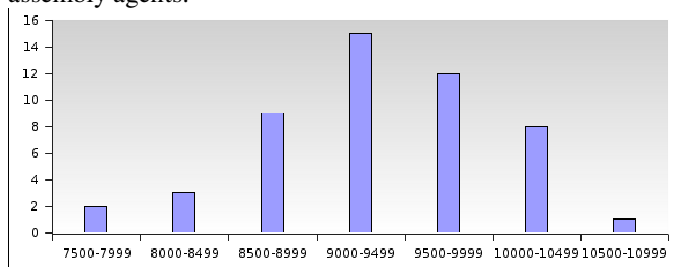


Figure 5

In our tests of the random method, we found that the behavior matched the expectations described in section V. Using a state machine with approximately 25% as many transitions as the guided method, the desired shape is built, filled in, and automatically repaired if it is damaged. As expected, it takes a long time for the shape to be filled nearly fully with assembly agents.

Fig. 6 displays a random assembly process in our simulator in several stages, (A) as a shell with a slightly higher density of assembly agents inside than outside, (B) as a somewhat packed shell which has had one corner destroyed, (C) after the corner is mostly repaired, and (D) with the

corner fully repaired, and the internal agent density significantly increased. As before, the black represents the obstacles and the grey represents the assembly agents.

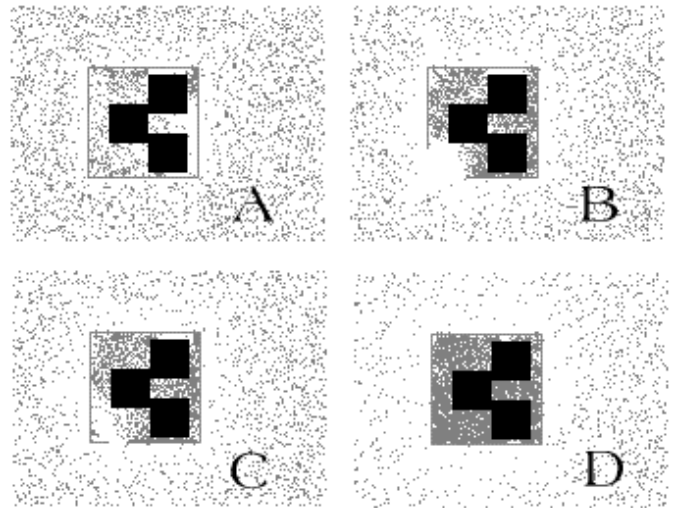


Figure 6

CONCLUSION

We have shown two methods by which an active self-assembly process can transform an object of unknown shape into an object of known shape by surrounding the unknown object with assembly agents. We have shown that these techniques work in large-scale simulations, and we have characterized their behavior analytically and empirically. We have shown that these techniques are flexible in terms of the shapes that can construct, and demonstrated self-repair using one of the techniques.

References

- [1] D. Ar Buckley and A. A. G. Requicha, “Active Self-Assembly”, *Proc. IEEE Intl. Conf. on Robotics and Automation*, New Orleans, LA, pp. 896-901, Apr 26 – May 1, 2004.
- [2] Chris V. Jones and Maja J. Mataric. "From Local to Global Behavior in Intelligent Self-Assembly". *Proc. IEEE Intl. Conf. on Robotics and Automation*, Taipei, Taiwan, pp. 721-726, Sep 14-19 2003
- [3] Wei-Min Shen, Peter Will, Berok Khoshnevis, “Self-Assembly in Space via Self-Reconfigurable Robots”, *Proc. IEEE Intl. Conf. on Robotics & Automation*, Taipei, Taiwan, pp. 721-726, Sep 14-19 2003.
- [4] Eric Klavins, Robert Ghrist, David Lipsky, “Graph Grammars for Self-Assembling Robotic Systems”, *Proc. Intl. Conf. On Robotics and Automation*, New Orleans, LA, pp. 5293-5300, Apr 26 – May 1, 2004
- [5] Shen, W.-M., Y. Lu and P. Will, "Hormone-based control for self-reconfigurable robots.", *Proc. Intl. Conf. Autonomous Agents*, Barcelona, Spain, pp. 1-8, June 3-7 2000.
- [6] B. Salemi, W.-M. Shen and P. Will, "Hormone Controlled Metamorphic Robots", *Proc. IEEE Intl. Conf. on Robotics and Automation*, Seoul, Korea, Vol. 4, pp. 4194-4199, May 21-26 2001.
- [7] H. C. Berg, *Random Walks in Biology*, Princeton NJ: Princeton University Press, 1993