Swarm-Mediated Cluster-Based Construction

S. Kazadi
J. Wigglesworth
A. Grosz
A. Lim
D. Vitullo
Jisan Research Institute,
28 North Oak Avenue,
Pasadena, CA 91107

The use of swarm clustering algorithms is investigated in the design of simple robots capable of carrying out swarm-mediated construction. Methods for generating multiple clusters of predetermined size are developed. Relative cluster motion algorithms are also developed and explored. All robotic algorithms are predicated on the use of robots that utilize no processing, global positioning, or explicit communication. Simple stigmergic communication and minimal sensing capabilities are used exclusively. Swarms of minimal agents are used to build equilateral triangles, squares, and pentagons. Future use of these methods in the design of more sophisticated construction techniques is discussed.

1. Introduction

Swarms have been the subject of increasing interest in recent years both as a result of their applicability to a wide range of different computational and physical problems, and as a result of the usefulness of generating robust algorithms using extremely simple agents. Agents in swarms under contemporary investigation need not have sophisticated models of the world, nor have any great capabilities, and yet the swarm itself can exhibit remarkably dynamic properties.

Many of the contemporary uses of swarms are motivated by biological systems. Among these are puck clustering system studies. These studies came about as a result of a desire to experiment with behaviors originally discovered in cemetary ants [1]. Cemetary ants have the remarkable ability to move dead carcasses seemingly aimlessly around an enclosed area, and end up with the dead ants all neatly piled in a single pile in some part of the enclosure. What is remarkable about this feat is that it happens with no central planning, no global knowledge; and, as far as can be discerned, no significant higher planning or thought in any of the agents.

Other researchers, including Holland and Melhuish [2] extended this work by building simple autonomous robots capable of carrying out a

similar task. The task, called *puck clustering*, consists of moving cylindrical objects, known as pucks, in an enclosed arena to a single pile. While the task itself sounds simple, the approach is minimalist, which means that the simplest possible robots must be used in order to create the cluster. The robots then built were extremely simple, bereft of any processing, global information, or communication (other than stigmergic communication). Yet despite these impediments, as with the ants, the robot group was able to move the pucks into a single central cluster.

These studies are centered on the development of swarm behaviors using a bottom-up approach. Such an approach typically designs the swarm by creating the individual behavior set, and altering the set based on its ability (or lack thereof) to complete a specific task. For example, Beckers et al. [3] generated an autonomous swarm that carried out clustering, but this study lacked any significant theoretical basis. In [4] and [5], two swarms were built which exhibited interesting properties. In the first, Winfield et al. [4] designed a swarm which communicated through a wireless LAN. In the second, Ijspeert et al. [5] implemented a swarm in which individual agents collaborate to lift sticks out of holes in the ground. In both cases, the swarms are capable of carrying out tasks requiring individuals to cooperate. This is an interesting and important part of swarm-based engineering. However, neither study yielded clues as to how a more general swarm-based methodology might be constructed. Rather, both studies were largely anecdotal, providing information about the design of specific swarms.

Kazadi et al. [6] provided an alternative way of approaching the problem. This approach examined the minimal conditions for clustering to occur, and generated robotic behaviors that could satisfy these requirements. The method was used to examine the dynamics of both an aphysical[1] simulated clustering system and a physical simulated clustering system.

To date, no practical use has arisen from clustering systems. However, recent developments have demonstrated the ability of swarms of minimalist robots to create a cluster containing all the pucks in an arena and theoretically determined how to minimize variance in the converging cluster size [7]. Kazadi et al. [8] illustrated the constuction of clusters of specific size and number. This capability suggests the possibility that the clusters can be placed in specific relative locations. Kazadi [9] further investigates this and discusses the methods in which single and multiple clusters can be formed, and how they can be moved relative to one another. Placement of clusters in specific relative locations is the first step

[1] An *aphysical system* is one in which the physics of the system is not a part of the simulation. Investigation is typically done using a simple probablistic model. The point of such studies is to examine the dynamics of a system without requiring the entire simulation, which generally produces numerical results quickly.

in cluster-based construction. By placing clusters this way, one can demarcate a construction task that can be based on these preliminary steps.

In this paper, we investigate the design of swarms of autonomous agents capable of generating groups of clusters and placing them at specific relative positions. Clusters generated by the agents consist of groups of differing constituent pucks placed at various relative positions. We proceed by initially creating a method of generating multiple clusters. Next, we explore a method of moving clusters that have formed. This leads to the design of a method for generating equilibria in which clusters move from initial locations to specific relative positions.

The remainder of this paper is organized as follows. Section 2 provides a review of theoretical clustering results drawn from earlier work. Section 3 describes a computer simulation used to explore these clustering algorithms. Section 4 presents cluster movement protocols and illustrates their use on groups of clusters. Finally, section 5 offers some concluding remarks.

2. The formation of single and multiple clusters

In this section, we develop a theoretical basis for the formation of single and multiple clusters, and elaborate on the implementation of this theory in simulation. Much of the theoretical work has been done elsewhere [6, 8], and we focus here on the main results. These theoretical results are important in understanding much of the implementation that is discussed in section 3.

2.1 Generation of clusters

Initial studies of the formation and dynamics of single and multiple puck clusters have been made primarily in the past several years. These grew out of a need to determine the conditions under which a set of robots of arbitrary capability could create a single cluster from a number of clusters or randomly scattered materials [6].

In [6], the likelihood that a robot approaching a cluster of size N (where N represents the number of elements of building material in the cluster, or the number of pucks) would pick up a piece of building material is denoted by a function $f(N)$. Likewise, the likelihood that a robot approaching a cluster of size N would drop off a piece of building material is denoted by $h(N)$. It is demonstrated that the minimal condition for the generation of a single cluster is

$$\frac{d(g(N))}{dN} \equiv \frac{d}{dN}\left(\frac{f(N)}{h(N)}\right) < 0. \tag{1}$$

This result means that the ratio of the likelihood to pick up and the likelihood to drop off is strongly monotonically decreasing. What this does

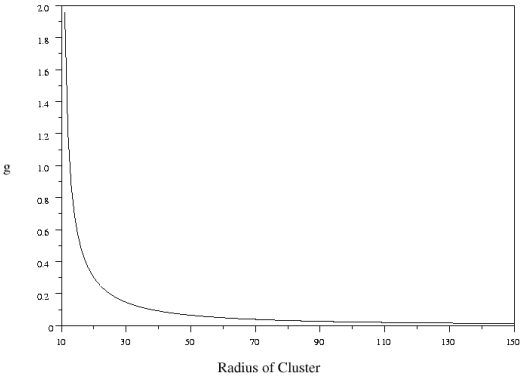

Figure 1. A form of the g functional that is suitable[2] for constructing a single cluster. Robots employing a behavior which creates this g form will build a single cluster from any initial distribution.

not indicate is the method of generation of this likelihood, which makes it particularly easy to use as a design specification. The requirement can be satisfied by any means, including some intelligent planning, random behaviors, or any other method.

A suitable form for the g functional is given in Figure 1. This form results in the construction of a single cluster.

Multiple clusters may be created by generalizing this method. This issue has been studied in two different models. In the first model [6], the generation of multiple clusters is the result of a g functional for which

$$\frac{d(g(N))}{dN} \equiv \frac{d}{dN}\left(\frac{f(N)}{h(N)}\right) > 0. \tag{2}$$

Any form of g satisfying this condition leads to the generation of multiple clusters because it favors removing material from larger clusters and depositing it on smaller clusters. If the particular number of clusters is controlled, this breakup process continues until all of the clusters have a single size, within random variation. However, this result is accurate only if the number of clusters is tightly controlled. If the number of clusters varies, this result cannot hold, as the number of clusters can increase unboundedly to the size of the number of building material elements. Algorithms conforming to these constraints allow multiple clusters of predetermined size to be reliably formed.

Examination of the generation of multiple clusters in an embodied simulation [6] led to the determination of more general conditions un-

[2]Any monotonically decreasing g will eventually result in a single cluster, but, as described in [6], a lower curve will form a single cluster more quickly than a higher curve.

der which clustering occurs. In embodied simulations, one of which is described in detail in section 3, the number of clusters cannot be explicitly controlled. Random placements and accidental separations of material from clusters can cause the unintended creation of multiple clusters. This variation alone makes the design of new multiclustering techniques necessary.

This was solved by making a *multiphasic g functional* in which the functional was decreasing (clustering) at low numbers of pucks and increasing at higher numbers of pucks. The minimal point became the actual size of the cluster, within some random variation.

Kazadi et al. [6] also investigated ways of implementing this kind of function using a thresholding model in which the behavior of the robot is determined according to the perceived cluster size. That is, the likelihood of a robot not holding material to pick up a piece of building material when it encounters a cluster is determined by

$$f(x) = \begin{cases} 0 < p < 1, & x \le T_{\min}, \\ 0, & T_{\min} < x \le T_{\max}, \\ 1, & T_{\max} < x, \end{cases} \tag{3}$$

where x is the perceived cluster size, p is some probability of making an error, $T_{\min}$ is the lower clustering threshold, and $T_{\max}$ is the cluster size. The corresponding likelihood of dropping off a piece of building material is given by

$$h(x) = \begin{cases} 0 < 1 - p < 1, & x \le T_{\min}, \\ 1, & T_{\min} < x \le T_{\max}, \\ 0, & T_{\max} < x. \end{cases} \tag{4}$$

The smallest region is provided to allow pseudoclustering.[3] Using these behaviors, Kazadi and colleagues were able to demonstrate that a single cluster would always be formed when the perceived size remained below the maximum threshold $T_{\max}$ (the clustering condition was satisfied). Moreover, when clusters that were larger than $T_{\max}$ were built (either initially or through random cluster growth), they would be reduced in size to that determined by $T_{\max}$. Thus, this behavior could achieve single or multiple clusters of a single predetermined size.

2.2 The role of density

In embodied robot simulations or with real robots, building material and interactions between robots and clusters of building material cannot be

[3] *Pseudoclustering* is a process in which random drop off and pick up cause clusters to grow in size, but there is no driving force toward single cluster formation outside of random percolation. The expected time to generate a cluster of a given size is proportional to p^N where p is the probability of dropping off a puck when the cluster is below the threshold size $T_{\min}$ and N is the number of building material elements in the cluster.

modeled numerically. This is because the detailed geometry of the interactions is important in determining the dynamics of the system. Kazadi et al. [6] determined a method of investigating the detailed dynamics of a clustering system using densities of material-carrying robots. In these simulations, the details of robot behavior were wrapped up in the functionals $F(\delta(\vec{x}))$ and $H(\delta(\vec{x}))$, where $\delta(\vec{x})$ is the position-dependent density of puck carrying robots. A formal definition of these functions and their relation to G can be found in [6]. In this case F and H represent the likelihood of material pick up and drop off, as with their perfect information analogs. It was possible to show that when

$$\frac{d}{d\delta}G = \frac{d}{d\delta}\left(\frac{F}{H}\right) < 0 \tag{5}$$

(or when the ratio is monotonically decreasing in density) the system will yield a single cluster. The power of the density-based analysis derives from its ability to be used to understand how imperfect information may be incorporated into aggregation theory.

One important result of density theory, which will become instrumental in later work, is that density gradients lead to material transfer. That is, if the density of material-carrying robots is high in one part of the swarm and low in another part of the swarm, there will be a transfer of material between the two parts. Thus, if the robot behavior creates a high density of material carrying robots in one area, and creates a lower density in another area, the material will move from the high to the low density areas. This net transfer of pucks forms the basis for many later higher-order behaviors.

2.3 Control of cluster sizes

We can use densities to determine the dynamics between clusters. That is, the amount that cluster sizes will vary randomly can be examined using densities as a starting point. We have already noted that in [6] it was determined that clusters that cause large densities tend to become smaller on average over time. This is because the swarm tends to equalize the density of pucks throughout the swarm, and so will move pucks from high to low density areas. Thus, if a cluster has a size greater than the g-function's minimum size, it will tend to create a higher density than smaller clusters. This will eventually lead to a reduction in the cluster's size, and a set of clusters with the same size.

Suppose that N_1 is at the equilibrium point and N_2 is above the equilibrium point. The density of the pucks around a cluster of size N_1 is

$$\langle D\rangle_1 = D\int_0^\infty \frac{F(N)}{F(N)+H(N)}\rho_{N_1}(N)dN, \tag{6}$$

where D is the density of robots, $\rho_{N_1}(N)$ represents the probability

density that a cluster of size N_1 will be perceived as having size N, and F and H are as previously defined. Around the cluster of size N_2 the density is

$$\langle D \rangle_2 = D \int_0^\infty \frac{F(N)}{F(N) + H(N)} \rho_{N_2}(N) dN. \tag{7}$$

Since the first cluster is at the equilibrium point and smaller than the maximum size, and the robots are in the threshold model, equation (6) reduces to

$$\langle D \rangle_1 = D \int_0^{T_{\min}} \rho_{N_1}(N) dN, \tag{8}$$

while equation (7) becomes

$$\langle D \rangle_2 = D \left(\int_0^{T_{\min}} \rho_{N_2}(N) dN + \int_{T_{\max}}^{N_2} \rho_{N_2}(N) dN \right). \tag{9}$$

In general, a larger cluster will lose pucks to a smaller cluster if

$$\int_0^{T_{\min}} \rho_{N_2}(N) dN + \int_{T_{\max}}^{N_2} \rho_{N_2}(N) dN > \int_0^{T_{\min}} \rho_{N_1}(N) dN \tag{10}$$

assuming that the densities are equal. Rearranging, we obtain

$$\int_{T_{\max}}^{N_2} \rho_{N_2}(N) dN > \int_0^{T_{\min}} (\rho_{N_1}(N) - \rho_{N_2}(N)) dN. \tag{11}$$

This condition will be satisfied by some forms of the probability of correct (and incorrect) characterization, while others will not be able to satisfy this in general or precisely at the equilibrium point. The integrand on the right is expected to be positive, but there is no reason to believe that this must be the case. N_2 must be large enough so that the integral on the left side is larger than that on the right. The loss of this condition can cause an increase in the cluster size, which in turn can lead to a significant variance in cluster sizes, depending on how sharply peaked the integral is.

One strategy which guarantees that the cluster size is controlled is to reduce $T_{\min}$ to zero. In this case, the integral becomes

$$\int_{T_{\max}}^{N_2} \rho_{N_2}(N) dN > 0. \tag{12}$$

This will be satisfied no matter how little N_2 exceeds the desired size, making the variance minimal. The only way to reduce the variance further would then be to make more precise measurements of cluster sizes, which would in practice increase the likelihood of generating

nontrivial structures in the clusters. This trade off is an important part of practical robot design.

2.4 Implementation issues

Equation (12) describes an important design paradigm in systems that use multiclusters. In multicluster systems, it is important to limit the variance in cluster size. Without this control over size, it is impossible for the system to accurately create intended structures. Thus, in all of our simulations, we limit the size variance by making the integral in equation (9) as close to zero as possible. This is accomplished by making $T_{\min}$ very close to zero, without being so close that agents will not pick up pucks from even the smallest clusters or that cluster shapes vary significantly from circular.

3. Simulation and sensory accuracy

In order to carry out our investigation, we utilize a computer simulation (Figure 2) of a physical clustering system, in which agents move and interact with other objects. We refer to these agents as *robots*. The other objects in the simulation are known as *pucks*, which are stationary, only moving when picked up or dropped off by robots. No other objects or agents exist in the simulation.

Due to our minimalistic approach, our robots are stripped of all advanced abilities and are left with extremely limited capabilities. They have the ability to perceive and navigate through the environment. They are equipped with grippers that can hold a single puck. They are bereft

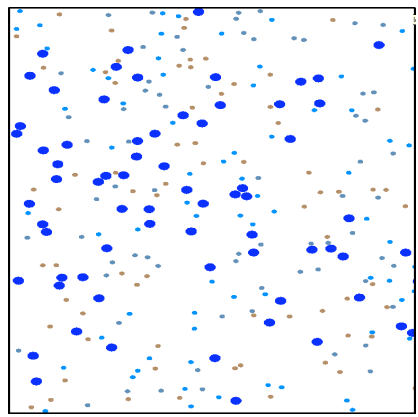

Figure 2. A screenshot of the simulation. The larger circles represent robots while the smaller circles represent pucks. 200 pucks and 50 robots are used with the robots and pucks having radii of 1 and 0.5 units respectively.

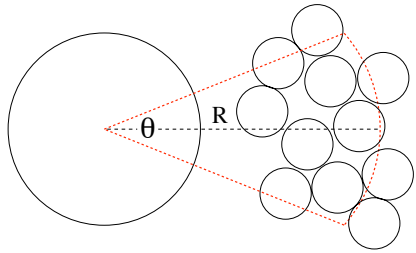

Figure 3. A robot approaching a cluster of building material. For simplicity, we assume that the cluster is circular. The robot will then estimate the cluster size by measuring what it perceives as the diameter of the cluster.

of any explicit communications and can only communicate with other robots through stigmergy.[4]

The simulation takes place in an unbounded two-dimensional plane. Although the simulation itself is only in two dimensions, real models will operate in three dimensions. Robots are randomly scattered throughout the arena. Pucks are placed in different ways, depending on the situation. To simulate a package drop, pucks may be placed in a specific location or locations of the arena. On the other hand, to simulate a cleanup operation, pucks may be randomly placed throughout the arena. The pucks can be defined to be of the same or different types.

In our simulations, robots have limited sensors, which are fixed in the robot's direction of motion. This sensor gives the robot the ability to perceive any objects in front of it and estimate the distance between the object and itself. Using this information, the robot can calculate the diameter of what is seen as a cluster inside its viewing angle (Figure 3).

The method of cluster size estimation, known as the *delta estimation method* [10], assumes that the width of the robot's viewing angle is as small as possible. As the robot approaches a cluster, it will try to estimate the cluster size by measuring the length of what is perceived to be the diameter of the cluster. This method is also responsible for maintaining the circularity of the cluster. This is true because any deviation from a circular structure forces the robot to regard the cluster as a smaller cluster from multiple directions. Any aggregate behavior, either adding pucks to or removing them from smaller clusters, will result in a circular cluster.

In what follows, we assume that the probability of approaching the cluster from any given angle is equal. Let L be the length of a chord through the cluster of radius R along the robot's direction of motion.

[4] *Stigmergy* is a method of communication in which environmental changes are used to pass information. By moving pucks to different locations, robots are able to leave primitive messages to other robots.

Then let α be the angle of approach, and

$$\alpha = f(L). \tag{13}$$

In our model, this indicates that

$$f(L) = \sin^{-1}\left(\frac{R}{R+r}\sin\left(\cos^{-1}\left(\frac{L}{2R}\right)\right)\right), \tag{14}$$

where r is the radial size of the robot. If $\rho(\alpha)$ is the probability of the robot approaching from any angle, then it is the case that

$$1 = \int_{\alpha_{\min}}^{\alpha_{\max}} \rho(\alpha)d\alpha = \int_{\alpha_{\min}}^{\alpha_{\max}} \frac{1}{\Delta\alpha}d\alpha, \tag{15}$$

where $\Delta\alpha = \alpha_{\max} - \alpha_{\min}$. Now suppose that

$$p = \frac{L}{2R}. \tag{16}$$

This defines a new function of p

$$F(p) = \sin^{-1}\left(\frac{R}{R+r}\sin(\cos^{-1}(p))\right), \tag{17}$$

where p runs from zero to one. Substitution and rearrangement gives us

$$1 = \int_{\alpha_{\min}}^{\alpha_{\max}} \frac{1}{\Delta\alpha}d\alpha = -\int_0^1 \left(\frac{1}{\Delta\alpha}\frac{dF}{dp}\right)dp. \tag{18}$$

Therefore,

$$\rho(p) = -\frac{1}{\Delta\alpha}\frac{dF}{dp} = \frac{R}{\Delta\alpha(R+r)\sqrt{(1-p^2) - \frac{R^2(1-p^2)^2}{(R+r)^2}}}. \tag{19}$$

If $R \gg r$, this reduces to

$$\rho(p) = \frac{1}{\Delta\alpha\sqrt{1-p^2}}. \tag{20}$$

With this probability density, and $\Delta\alpha = \pi/2$,

$$\langle p\rangle = \frac{2}{\pi}\int_0^1 \frac{p}{\sqrt{1-p^2}}dp = -\frac{2}{\pi}\sqrt{1-p^2}|_0^1 = \frac{2}{\pi}, \tag{21}$$

which gives

$$\langle L\rangle = 1.2732395R. \tag{22}$$

As an example, if we assume the radius of cluster R is 1 and the radius of a robot is 0.1 then

$$\rho(p) = \frac{1}{1.1 \sin^{-1}\left(\frac{1}{1.1}\right)\sqrt{(1-p^2) - \frac{(1-p^2)^2}{1.1^2}}}. \tag{23}$$

This gives us the probability density of the robot estimating that a cluster has a particular ratio of the specified cluster size. In addition, equation (19) is nearly constant for clusters whose radius is much greater than that of the robot. This indicates that the estimated size should be linearly dependent upon the actual size of the cluster.

4. Movement of clusters and formation of shapes

In puck clustering systems, the final position of a cluster cannot be determined before it is completely formed. Thus, the need arises to relocate a cluster to its new position, after which construction can begin.

As shown in subsection 2.2, the equilibrium density of puck carrying robots is very closely tied to the behavior of robots in the presence of pucks. This equilibrium density will be maintained by the dynamics of a clustering system so that variation in robot density will return to the equilibrium value.

This equilibrium behavior can be exploited so as to cause a creeping motion of the cluster in any given direction. This situation is illustrated in Figure 4. The physics relates to the diffusion of materials in a closed environment; in diffusive systems, high density regions spread to low density regions. Dynamic equilibria of differential equilibrium generators perform in the same way as this system.

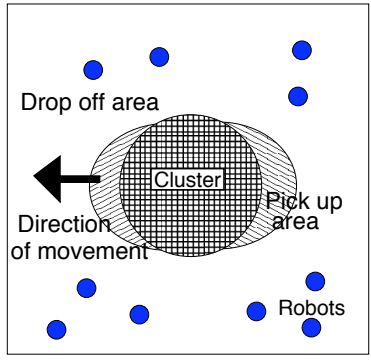

Figure 4. A movement-generating differential. The areas to the right and left of the cluster represent areas where robots would be more likely to pick up rather than drop off a puck or *vice versa*.

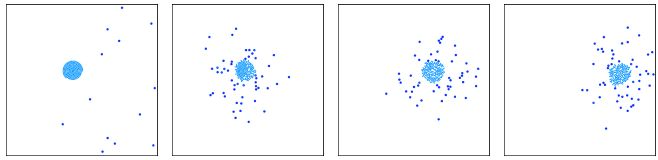

Figure 5. The movement of a cluster. 200 pucks and 50 robots are used with the robots and pucks having radii of 1 and 0.5 units respectively. Information on the specific f, h, and g functions used can be found in [7].

4.1 Linear cluster motion

The simplest type of motion is linear. In this mode, the cluster moves directly in a predefined direction ϕ with a desired speed. This can be accomplished by the method described, if the two regions are on directly opposite sides of the cluster. We utilize the following procedure in our simulations.

First, we denote the likelihood that a robot approaching a cluster of size N will remove a puck from the cluster by $f(N)$. We likewise denote the likelihood that a robot will drop a carried puck onto a cluster of size N by $h(N)$. We can achieve linear locomotion by influencing these probabilities. Let us denote the probability of a robot facing direction θ picking up a puck P_u as

$$P_u = f(N)((1-\epsilon) + \epsilon\cos(\theta - \phi)) \tag{24}$$

and the robot's probability of dropping a puck in the cluster P_d by

$$P_d = h(N)((1-\epsilon) - \epsilon\cos(\theta - \phi)). \tag{25}$$

These probabilities induce a higher carrier density at the angle $\phi - \pi$, and a lower density at the angle ϕ. As a result, we can expect this behavior to lead to the movement of pucks from the cluster side at angle $\phi - \pi$ to the cluster side at angle ϕ. This results in a motion toward the angle ϕ. In these equations, ϵ is a constant and controls the speed at which the cluster travels as a result of the size of the perturbation of basic robot behavior. Applying this to our simulations, we can create locomotion of the cluster (Figure 5).

We can derive the speed of the cluster as a function of ϵ by considering the rate of change of the number of pucks on either side of the cluster. First, let us consider the following expression for dN_i/dt, the rate of change in the number of pucks in cluster i:

$$\frac{dN_i}{dt} = \alpha(-(r_t - r_c)P_u + r_c P_d), \tag{26}$$

where r_t represents the total number of robots and r_c the number of robots that are carrying pucks. We assume that a robot density of α

is constant around this cluster. By substituting P_u and P_d as in equations (24) and (25) and integrating, we get:

$$\int_{LB}^{UB} \frac{dN_i}{dt} d\theta = -2\alpha\epsilon r_t \sin(\theta - \phi) \frac{f(N)h(N)}{f(N) + h(N)} \Bigg|_{LB}^{UB}, \tag{27}$$

where UB is the upper integration bound and LB is the lower integration bound.

In order to get an idea of how quickly this cluster is moving, we will measure the number of pucks being transferred to and away from each side of the cluster. The actual value may be obtained by evaluating the integral using the two intervals $[\phi - \pi/2, \phi + \pi/2]$ and $[\phi + \pi/2, \phi + 3\pi/2]$. For the interval $[\phi - \pi/2, \phi + \pi/2]$, we obtain

$$\int_{\phi-\pi/2}^{\phi+\pi/2} \frac{dn_i}{dt} d\theta = 4\alpha\epsilon r_t \frac{f(N)h(N)}{f(N) + h(N)} \tag{28}$$

and for the interval $[\phi + \pi/2, \phi + 3\pi/2]$, we obtain

$$\int_{\phi+\pi/2}^{\phi+3\pi/2} \frac{dn_i}{dt} d\theta = -4\alpha\epsilon r_t \frac{f(N)h(N)}{f(N) + h(N)}. \tag{29}$$

In other words, pucks are being accumulated in the interval of the desired direction of cluster motion while they are being removed from the opposite interval. As a result, pick up and drop off differentials exist, which result in the movement of the cluster towards the drop off differential.

We can estimate the speed at which the cluster moves by examining the cross section of the cluster as it moves. If we estimate that the cross section of the cluster is $\pi R = \pi d_p \sqrt{N}$, where d_p is the diameter of a puck, and the speed of the cluster is given by v_l, the rate of increase in area is given by

$$\pi d_p \sqrt{N} v_l = \int_{\phi+\pi/2}^{\phi+3\pi/2} \frac{dn_i}{dt} d\theta = -4\alpha\epsilon r_t \frac{f(N)h(N)}{f(N) + h(N)}, \tag{30}$$

which gives us

$$v_l = \frac{\epsilon 4\alpha r_t}{\pi d_p \sqrt{N}} \frac{f(N)h(N)}{f(N) + h(N)}. \tag{31}$$

Clearly, this is a linear function of the epsilon, which gives us a convenient way of controlling the speed. The maximum speed is given by

$$v_{\max} = \frac{4\alpha r_t}{\pi d_p \sqrt{N}} \frac{f(N)h(N)}{f(N) + h(N)} \tag{32}$$

for any given cluster size N.

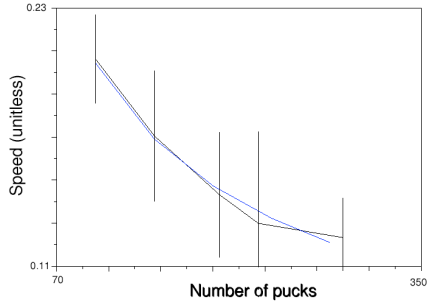

Figure 6. The cluster speed as a function of its size, holding the density of robots α constant. The predicted behavior of the cluster speed matches well with the observed behavior.

If N is very large then $f(N) \ll h(N)$. Thus,

$$v_{\max} \approx \frac{4\alpha r_t}{\pi d_p \sqrt{N}} f(N), \tag{33}$$

which is inversely proportional to the square root of the number of pucks. Figure 6 illustrates how the cluster speed varies as its size increases. This movement has very good agreement with the expected behavior.

4.2 Relative cluster motion

Now that we have seen how cluster motion may be created, we must examine how to determine its direction. In subsection 4.1, we assumed that the direction ϕ was the direction of motion. How is this ϕ determined?

4.2.1 Pairwise cluster positioning theoretical considerations

In order to answer this question, we look at the overall goal of this line of inquiry. We are interested in placing clusters at precise relative distances from one another. What we are not interested in, however, is the precise placement of clusters at specific positions in any particular area. That is, we are not investigating methods of placing cluster A at position $(3.12, 4.25)$ in some coordinate system with other clusters in specific relative positions and orientations.

These considerations lead us to the conclusion that our desired direction of motion is that direction which moves the clusters toward satisfying the desired set of distance constraints. For example, let us suppose that we have two clusters in our system. We may choose as a direction a vector lying on the connector of the centers of the two clusters. This will move the clusters monotonically towards or away from one another, irrespective of their initial positions.

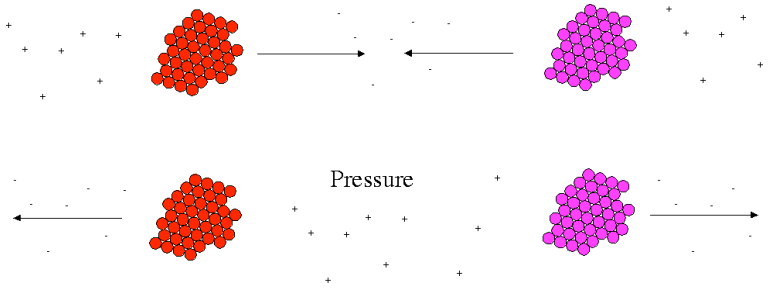

Figure 7. The variation of puck carrying robot density in two different actions. Decreasing the inside density of puck carrying agents tends to move clusters together (top) while increasing the inside density tends to move the clusters apart (bottom).

How might this be accomplished? We demonstrated in sections 2 and 3 that the density of puck carrying robots is related to the dynamic properties of the clusters, and that this density may be exploited in order to create cluster movement in any given desired direction. Thus, we may use a behavior in which these densities are increased directly between the robots and decreased on the outside of the clusters. The situation is depicted in Figure 7. Once the clusters are at the right distance, the density differential disappears, leaving the clusters at rest.[5]

4.2.2 Design of agents for pairwise cluster positioning

The agents used in our simulations are meant to be simple. As a result, we use a simple thresholding behavior for the robots. They carry out this behavior in order to decide whether or not to pick up or drop off a puck when encountering a cluster. This behavior, already explored in section 2, allows the agent or agents to create individual or multiple clusters. Whether or not the system develops multiple clusters depends on whether or not the upper threshold is greater than the number of pucks in the system.

This simple behavior may be perturbed in order to create a density differential as indicated in Figure 7. We accomplish this by changing the pick up and drop off probabilities from

$$P_u = \Phi(T_l - p) + \Phi(p - T_u), \tag{34}$$

$$P_d = \Phi(p - T_l) - \Phi(p - T_u), \tag{35}$$

[5]In practice, the creation of such an algorithm requires as a prerequisite the use of a sensor capable of identifying puck clusters and of accurately identifying their distance. Such a sensor would need to be placed above the robot's main mass in order to be able to see pucks quite a distance away, and behind other pucks and robots. This would seem to be a constraint that would impact the design of a physical robot capable of implementing these algorithms.

where p represents the number of pucks in the cluster, T_l represents the lower bound, T_u represents the upper bound, Φ is a step function

$$\Phi(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}, \tag{36}$$

to

$$P_u = \Phi(T_l\Pi(d, D, n_c) - p) + \Phi(p - T_u), \tag{37}$$

$$P_d = \Phi(p - T_l\Pi(d, D, n_c)) - \Phi(p - T_u), \tag{38}$$

where $\Pi(x, y, z)$ is defined by

$$\Pi(x, y, z) = \varrho(x, y)(\Phi(x - y)\Phi(z - 1) - \Phi(y - z)\Phi(2 - z)), \tag{39}$$

and $\varrho(x, y)$ is a function such that

$$\frac{d\varrho}{d(x - y)} > 0. \tag{40}$$

As an example, we use

$$\Pi(x, y, z) = (x - y)(\Phi(x - y)\Phi(z - 1) - \Phi(y - z)\Phi(2 - z)) \tag{41}$$

in our simulations.

Let us examine the effect of this behavior on the clustering system. Recall that, using our definitions, we have

$$\Pi(d, D, n_c) = (D - d)\Phi(n_c - 1). \tag{42}$$

The method is used to change the lower bound of the clusters so as to move the clusters either towards or away from one another. If the distance between the clusters d is greater than the desired distance D, the change in desired cluster size is negative, but only on the side of one cluster furthest away from the other cluster. This will tend to move the clusters towards one another. On the other hand, if the current distance is smaller than the desired distance, the change is inverted, and the clusters move apart. Finally, since the amount of change is proportional to the distance between clusters, this will result in the speed of the motion proportional to the error in the desired versus the actual distances. The resultant behavior is shown in Figure 8.

4.2.3 Generalizations for multiple pucks

The method used to direct the motion of the clusters discussed in subsection 4.2.2 cannot be generalized to generate precise positioning of multiple clusters. This is because it assumes that all clusters are meant to be a distance D from one another, and thereby leads to the creation of multiple cluster formations containing identical cluster numbers. The variablility in the final arrangement makes it impossible to generate arrangements of specific design, particularly if the desired design is not symmetric.

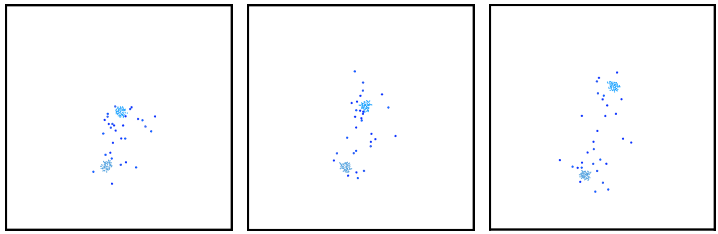

Figure 8. The use of the motion algorithm given in equations (41) and (42) to move two clusters apart to a predetermined relative distance. 100 pucks and 50 robots are used with the robots and pucks having radii of 1 and 0.5 units respectively. Information on the specific f, h, and g functions used can be found in [7].

In order to place the clusters precisely, symmetry must be avoided. This means that the building activities of the robots cannot be uniform. That is, some inhomogeneity, either by information about the relative placement of all clusters or by cues found in the clusters themselves must be utilized to break symmetry. Methods that utilize the relative positons of all clusters require a great deal of processing, either from a global surveyor or independently, with each individual robot having the ability to accurately map the building arena. Such a method can only work in situations where the entire arena can be surveyed, which may not necessarily be the case. In such an occasion, the method will fail if the system described is applied to a case where only some of this information is available. Thus, looking ahead to later studies, we choose to bypass the first option and pursue the second.

Thus, we choose to break symmetry by providing cues in the system. This can be accomplished by "coloring" the pucks so that it is possible to distinguish the different types of building materials. Once this is done, the agents are capable of making decisions about how to interact with groups of clusters based on the details of the relative positions of the different clusters and their types.[6]

Once the different building materials have been colored, it is important to separate them into individual clusters. Thus, a behavior implementing building material segregation must be adopted. The minimal condition leading to segregation of building material is that no material be placed down next to material of a different color. However this is achieved, the outcome will be the eventual segregation of all materials randomly distributed. Moreover, if the material is distributed nonrandomly, placing them in randomly created clusters of uniform

[6] Of importance is the relation of this latest approach to the *minimal condition* required for correct placement of the clusters. This minimal condition is that the number of desired clusters emerge and monotonically approach the desired organization of clusters.

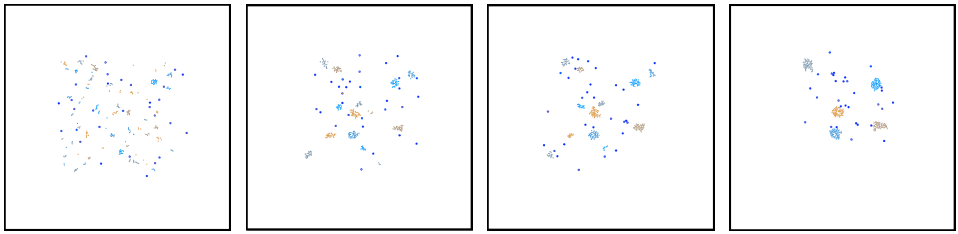

Figure 9. This figure illustrates the segregation process of a number of initially scattered clusters using robots that randomly move around, picking up and depositing pucks according to the clustering criterion given above, and restricting deposition to clusters containing only one identical puck type. 500 pucks and 30 robots are used with the robots and pucks having radii of 1 and 0.5 units respectively. Information on the specific f, h, and g functions used can be found in [7].

type outside of the original cluster arrangements will lead to the generation of clusters of uniform color. The situation is depicted in Figure 9. Clusters segregated by color may now be relatively placed according to their desired pairwise difference.

In order to build robots that move a cluster in an appropriate direction depending on the pairwise distance of each cluster, each robot must have data indicating the desired distance between clusters of some puck type and the clusters of every other puck type. This would seem to be a minimal requirement of any system in which the robots are expected to be able to properly place the clusters, irrespective of the cluster identification method. Moreover, in order for this information to be effective, the robot must have the ability to determine the pairwise distances between clusters that are nearby.[7] This requires that the robot have the ability to sense the approximate centroid of the cluster, and calculate the distances between clusters of differing types. We thus assume that all robots have these capabilities, which serve as minimal engineering constraints for robots capable of carrying out this type of construction.

In order to move clusters in a direction that will provide correct placement, we must generate a direction of motion that minimizes the difference between the desired and the current relative position. The difference between the current relative position and the desired relative position is given by

$$d = \sum_{i \neq j} (d_{ij} - D_{ij})^2, \tag{43}$$

where d_{ij} is the actual distance between clusters i and j, and D_{ij} is the

[7]For the purposes of this study, we assume that all clusters are nearby.

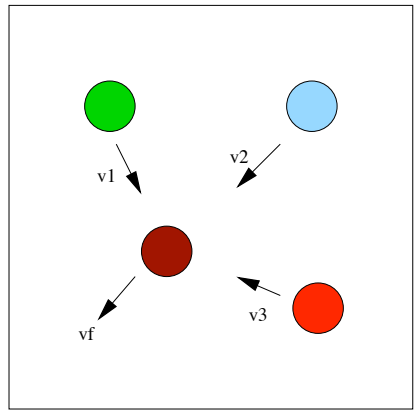

Figure 10. This figure illustrates the contributions to the motion of a single cluster. The contribution of each pair is aligned with the vector joining the centers of the two clusters. The sum of the vectors determines the overall direction of motion and its magnitude (v_f).

desired distance between these clusters. If we take the variation of this distance as

$$\vec{\delta d} = 2 \sum_{i \neq j} \widehat{u_{ij}} (d_{ij} - D_{ij}) \tag{44}$$

where the direction $\widehat{u_{ij}}$ is taken to be along the connector between cluster i and cluster j. The magnitude of $\vec{\delta d}$ is proportional to the amount that the upper bound of the cluster in the thresholding model in equations (41) and (42) will change by. Notably, this goes to zero when all the distances are as desired, and has a difference in sign depending on the deviation from the desired distance. The method is depicted in Figure 10.

A word about the methodology is appropriate here. The application of swarm engineering to this problem was ostensibly centered around the generation of minimal conditions required for the generation and movement of clusters of materials initially scattered or in a pile presumably already built. However, the past two subsections have laid out a methodology rather than focusing on the nature of swarm engineering itself. This would seem contrary to the methodology we are adhering to in this study.

However, an examination of the past two subsections has indicated that a careful adherence to the methodology has indeed occurred. This is because the minimal condition for the generation of clusters was first identified. Then, the minimal condition for the movement of clusters (along with a methodology for achieving it) was presented. Finally, a minimal condition for applying this to irregular shapes was explored and

a process was chosen. This process itself required a minimal condition for segregating pucks, which was derived along with a methodology for carrying this out.

What we have not done is develop a minimal condition that guarantees that the method will always move clusters into the correct final position, irrespective of their initial positions. In fact, as we shall see, our failure to guarantee this last condition makes it possible for the final configuration to have many different final organizations.

4.3 Generation of simple geometric shapes

We have examined the design of a system ostensibly capable of generating a predetermined number of clusters that are in predetermined relative positions. In this subsection, we apply the methodology to a system made up initially of scattered building material that must be clustered and positioned as a precursor to construction. As we are particularly interested in issues related to the construction of structures based on this system, we examine the final design in terms of its deviation from the predetermined structure.

We initialize our simulations with 50 robots and 100 pucks per puck type. As we shall see, the methodology developed thus far is capable of producing desired structures, but it is also plagued with the generation of *degenerate structures*, which we will explain more fully. As a result, we limit ourselves to structures containing three, four, and five clusters. All experiments begin with all building material and agents randomly scattered throughout the arena. No agents initially carry pucks, so they must carry out foraging and collection activities.

Although the method may be used to build asymmetric structures, Figure 11 presents simulations in which symmetric structures of three, four, and five clusters are formed. The clusters in these example construction tasks form over many iterations of the pick up/drop off behavior, and take their places in the vertices of the preplanned symmetric structures. In these cases we formed equilateral triangles, squares, and pentagons.

When building these structures, it is interesting to ask what the specificity of the construction task is. That is, how closely does the final structure match the desired structure? In order to determine this, we measured the deviation over several iterations of the final design once the structure had been formed. The distances were measured from center to center of the clusters and compared to the desired values. The results are summarized in Table 1.

While only three structures have been represented here, the results illustrate a striking specificity for the equilateral triangle and square. However, the pentagon has a very large deviation. This lack of specificity may be the result of the small distances used, compared with the size

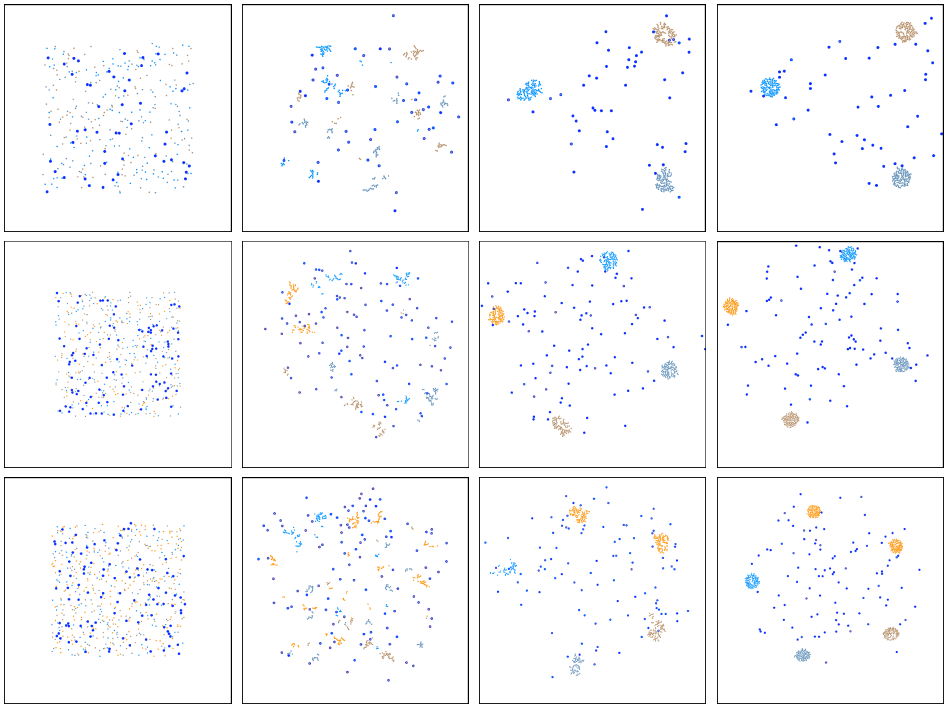

Figure 11. The spatial changes of the organization of building material under the action of clustering robots carrying out cluster placement in the correct placement multiple clusters. Initially, the clusters are poorly defined due to the effect of cluster movement, which tends to split up formed clusters. However, once the majority of building materials of the correct types have been placed in the appropriate position, or nearly so, the clusters tend to form as circular metaobjects. The final placement is indicative of the type of specificity that is possible in clustering construction. For the triangle, square, and pentagon simulations 300 pucks and 50 robots, 400 pucks and 100 robots, and 500 pucks and 150 robots are used respectively. All the robots and pucks have radii of 1 and 0.5 units respectively. Information on the specific f, h, and g functions used can be found in [7].

Structure	Deviation
Equilateral triangle	3.392%
Square	1.778%
Pentagon	15.062%

Table 1. The percentage deviation of distances between any two clusters in a given structure over the desired distance during many iterations of robot action after the construction has concluded. The results indicate a relatively good specificity for the simpler two structures with a poor specificity for the pentagon.

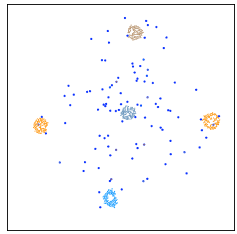

Figure 12. One degenerate form of a pentagon. Note that the positions are not as desired, though the clusters are in the final metaphysical form.

of the clusters. The deviation represents a rather small variation in the position of the cluster, which can be a large part of the structure's size. Moreover, the deviations can be reactive and additive. That is, a single variation in one cluster position can cause other clusters to move. This is akin to crystalline structure motion under heating. If the structure is too variable, then it is unstable. With both of these problems, however, larger structures with similar positional specificities may reduce both the overall effect of jiggling and the concommittant movement of other clusters due to the jiggling of nearby clusters.

A secondary, and significantly more important property of this type of clustering and structure construction is the existence of degeneracies. Degeneracies refer to secondary structures forming with the same or similar sets of distances. These form because the random initial placement of clusters creates cluster bundles which cannot be moved to the appropriate relative positions. This can happen because the initial position of the cluster is such that moving the cluster (in two dimensions) requires the movement of other clusters already placed. If these clusters are moved, then the concommittant extra deviation from the desired final cluster placement is so great that they tend to move back into place. That is, the moving cluster cannot "push" its way through. As a result, it gets stuck where it is, and the structure becomes permanent. The situation is depicted in Figure 12.

Several different forms can occur which become locked in for structures with more than three clusters. Clusters may initially form far from their desired locations. Other clusters may arrive at their own desired locations first, and then block the movements of clusters still out of position. The overall structure then takes on a configuration different from the desired one. This can be thought of as being analogous to the difficulty of changing the structure of a sterically blocked chemical structure at low temperature.

We report the percentage of runs out of 100 which develop the correct structure in Table 2. As can be seen, as the number of clusters increases, the number of degenerate forms increases from zero in the case of the

Structure	Correct shape percentage
Equilateral triangle	100%
Square	66.667%
Pentagon	12.5%

Table 2. The percentage of completion of the correct shape for each of the three shapes discussed.

equilateral triangle to seven out of eight final forms. This cannot be overcome using the current scheme due to the random initial positions of the clusters and the movement methodology.

Thus, the methodology we have examined is capable of building clusters, moving them relative to one another, and positioning them rather precisely relative to one another. However, the difficulties associated with this methodology need to be handled in order to make it reliable.

5. Discussion and concluding remarks

Clustering is historically the basis for the development of construction. Throughout human history, large complex objects have been built from small, simple objects. The methodologies have been varied and have utilized increasing sophistication. Most of the technologies utilize some type of glue or fastening device that holds the building blocks together. These make up the bulk of the construction techniques used on Earth. However, the most elegant solutions seem to have come from the use of clever design, such as those of Renaissance cathedrals or Incan temples. These latter designs seem to have depended very much on the structure and shape of the building materials themselves and less on fastening with mortar.

In all construction tasks, the development of an accurate measurement and marking technique is an important first step. Normally, this is accomplished by making careful measurements of all relevant sizes of the constituent parts of the overall construction. Moreover, careful placement of these parts using sophisticated equipment and processes is part of the complete job. All of the techniques in use today require a careful understanding of the job that is being done, the method being used, and the sequence of steps required.

When using multiple robots, this type of understanding cannot be utilized. There is no way to communicate to contemporary artificial agents the meaning behind what they are doing nor expect them to be able to adapt to any changing environment so as to compensate for errors. Thus, our mechanisms for doing what seem to be some of the most trivial tasks are significantly more cumbersome than the same tasks might be if given to an alert craftsman.

However, the most difficult step in our construction task of single-level three-dimensional structures would seem to be marking the boundaries. This is because the swarms cannot be assumed to have access to global data which would simplify the task. The application of building material to the top of the cluster (so as to build columns) or to the space between clusters (so as to build walls) seems, without the benefit of hindsight which will surely come later, much simpler than the initial marking of all important positions by clusters.

What we have developed is a method of constructing multiple clusters and moving them in controllable ways into exact locations relative to one another. This methodology, or a derivative, would seem to be able to accomplish what are probably the most difficult tasks in the autonomous construction of single-level structures: correctly marking the area for later excavation, laying the foundation, and raising walls and ceilings.

The problem with the current methodology is that it is not capable of uniquely producing desired two-dimensional structures, but rather produces a number of different structures including the desired one. The increasing number of degenerate states as the number of clusters increases seems to be a significant problem. Future work will have to find appropriate ways of modifying the current strategy so as to allow the degenerate states to disappear, leaving only the single desired final state.

However, despite this problem, the current work represents a significant step forward in this field. Previous work has concentrated only on the development of one cluster or multiple (predetermined) clusters using a swarm protocol. This work concentrates on the development of practical applications of the swarm-based clustering technique which requires little more sensory or cognitive capability than those used in simple clustering robots.

This sets the stage for significant further work in this area. While it is speculative at present, one might imagine the degeneracy problem being solved in a year's time, leading to the development of nondegenerate cluster-based structures. Moreover, while this work is presently accomplished only in simulation, it is realistic to believe that it can be the precursor of real clustering work with robots. Finally, though this is highly speculative, one might imagine the first single-level three-dimensional structures being built by swarms of simple robots within five years. The possibilities once this step has been accomplished include the automated construction of commercial, residential, and pioneering structures in local and remote locations.

6. Acknowledgments

J. Wigglesworth and A. Lim gratefully acknowledge support by the Center for Neuromorphic Engineering at the California Institute of Tech-

nology, made possible as part of the Engineering Research Center Grant Number EEC-0313658. The authors would also like to thank O. Koroleva for valuable comments and discussion of the issues presented herein.

References

[1] J. Deneubourg and S. Goss, "Collective Patterns and Decision-making," *Ethology, Ecology, Evolution*, **1** (1989) 295–311.

[2] O. Holland and C. Melhuish, "Stigmergy, Self-organization, and Sorting in Collective Robotics," *Artificial Life*, **5**(2) (1999) 173–202.

[3] R. Beckers, O. Holland, and J. Deneubourg, "From Local Actions to Global Tasks: Stigmergy and Collective Robotics," in *Prerational Intelligence: Adaptive Behavior and Intelligent Systems without Symbols and Logic*, volume 2, edited by H. Ritter, H. Cruse, and J. Dean (Kluwer Academic Publishers, 2000).

[4] A. Winfield and O. Holland, "The Application of Wireless Local Area Network Technology to the Control of Mobile Robots," *Microprocessors and Microsystems*, **23**(10) (2000) 597–607.

[5] A. Ijspeert, A. Martinoli, A. Billard, and L. Gambardella, "Collaboration through the Exploitation of Local Interactions in Autonomous Collective Robotics: The Stick Pulling Experiment," *Autonomous Robots*, **11**(2) (2001) 149–171.

[6] S. Kazadi, A. Abdul-Khaliq, and R. Goodman, "On the Convergence of Puck Clustering Systems," *Robotics and Autonomous Systems*, **38**(2) (2002) 93–117.

[7] A. Zhang, M. Chung, B. Lee, R. Cho, S. Kazadi, and R. Vishwanath, "Variance in Converging Puck Cluster Sizes," in *Proceedings of AAMAS 2002*, edited by C. Castelfanchi and W. L. Johnson, Bologna, Italy, 2002.

[8] S. Kazadi, M. Chung, B. Lee, and R. Cho, "On the Dynamics of Clustering Systems," *Robotics and Autonomous Systems*, **46**(1) (2004) 1–27.

[9] S. Kazadi, "Position Control in Puck Clustering Systems," in *Proceedings of the Workshop on Agent and Swarm Programming, 2003*, edited by M. Kirschenbaum and D. Palmer, Clevelend Ohio, September, 2003.

[10] S. Kazadi, "Swarm Engineering," Ph.D. Thesis, California Institute of Technology, 2000.