Robot Clustering

C. Lee, M. Kim, S. Kazadi*
Jisan Research Institute
28 N. Oak Ave.
Pasadena, CA 91107

Abstract

Puck clustering systems are systems in which simple agents move building material, or pucks, in a spatially limited area in a random or pseudo-random way. While we adapt puck clustering theory to robot clustering systems to generate a decentralized swarm of robots which coalesces using only stigmergic information and local sensing into a single cluster, this paper does not discuss puck clustering. Rather, its focus is on aggregation. Robot clustering systems may be characterized by the number of active robots in the system and the average variance of the robots from a determined center. The number of active robots decreases as cluster is formed, mirroring the analogous result of puck clustering. There is a sharp decline in the average variance of the robots, indicating a rapid coalescence of the robot swarm.
keywords: puck clustering, robot clustering, swarm engineering

1 Introduction

For many years, the use of swarms and clustering systems has been examined as a potential solution to remote construction problems. Historically, clustering systems were inspired by biological swarms like those of ants and termites. In this study, however, we are interested in engineering a clustering system that satisfies specific characteristics. As a result, we are not interested in biological swarms [2, 7] and they will not be used to motivate our design. One of the methods that has been investigated in recent studies [1] is based on puck clustering. These studies have demonstrated how one might use puck clustering as a precursor to construction, with initial steps centered on clustering and correct cluster placement. The clustering process moves pucks into one or more clusters which are later moved to desired positions. The movement of the final clusters or clusters is generally necessary, as it is unlikely that they will emerge in predetermined positions. The main difficulty in moving clusters is that they cannot be easily destroyed and reconstructed at the final desired postions, as the swarms are typically decentralized and have no global information as to where the clusters should be placed. This would seem to be a generally easier task if the cluster were to stay intact as opposed to breaking up. Thus, in [1, 6] robots randomly and individually pick up pucks from one side of the cluster and relocate them to the opposite side of the same cluster. Repeated applications of this basic behavior move the clusters without changing their global structure, or relative placement.

The main problem with this method of cluster movement lies in the required time. A cluster moves excruciatingly slowly with respect to the robot's speed and would seem to be an impediment to realistic construction tasks. As a result, it is perhaps more desirable that robots arrange themselves into clusters and move the cluster or clusters using one of the many available flocking mechanisms [5, 9]. Flocking is the formation and maintenance of coherent group movement, implemented by a set of mobile agents. While using a flocking mechanism, there are chances of individual robots of the flock not sensing each other, or even of the flock bifurcating because of an obstacle or another source of interference. The broken flock would then need to somehow coalesce, which cannot be done easily. By utilizing clusters of robots rather

*To whom correspondences should be addressed at sanza@jisan.org.

than clusters of pucks, one can take advantage of the same properties of clustering that exist in puck clustering systems (robustness, simplicity, ease of control), while taking advantage of the flocking potential of clustered robots once the clusters have formed.

Built with the same theoretical concepts of puck clustering, robot clustering is able to perform the same type of task as puck clustering: aggregate robots without any prior knowledge of the global structure. Puck clustering occurs even if agents do not have knowledge of the global arrangements of pucks. We demonstrate that the same can be true for robot clustering even in an unbounded arena. Once clustering has been achieved, it may be directly applied to the first stage of construction: precise cluster placement.

The remainder of the paper is organized as follows. In section 2, we discuss some previous results from puck clustering of importance to this work. Section 3 explores how puck clustering may be adapted to robot clustering. Section 4 describes the results of our simulations of this work. Section 5 presents a discussion of these results and indicates future directions of this research.

2 Puck clustering

Recent studies on puck clustering have determined a number of important properties for clustering systems [1, 3, 4]. In this section, we review the most important results from these studies relevant to our work on robot clustering.

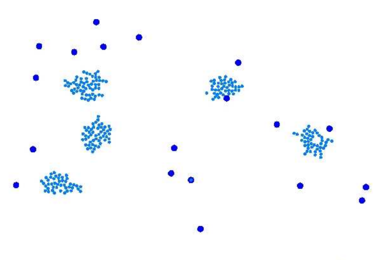

Figure 2.1: Above is a screen-shot of a simulation in which robots cluster pucks.

We may characterize puck clustering systems in the following way. Firstly, there are two different types of "things" in the system. The inactive objects, pucks, do not have any behavior, but may be picked up and put down. The active things in the systems are the agents, which have the ability to pick up pucks, move them about, and put them down. These agents also have enough sensory capability to be able to determine whether or not they are about to collide with an obstacle, and whether or not they are immediately next to a cluster. Moreover, we assume that they have the ability to characterize the cluster's size. The types of sensors that may be used by agents include video cameras and IR sensors, which can differentiate what is detected. The behaviors of agents are restricted to moving about, picking up pucks, and putting them down. A secondary restriction is that robots can only place pucks next to existing clusters. If a robot approaches a cluster, we define the likelihood that it will pick up a puck from the cluster with the function $f(N)$ where N is the number of pucks in the cluster. The likelihood that the robot will drop off a carried puck, on the other hand, may be represented by the function $h(N)$. The two functions, f and h are not correlated. For instance, if a robot always picks up pucks, $f(N) = 1$; however, this does not mean that $h(N) = 1$. Then, we define a function g by

$$g(N) = \frac{f(N)}{h(N)}. \tag{1}$$

In such a system, the necessary and sufficient condition for clustering to occur is that g be monotonically decreasing.

In [4], a secondary criteria based on densities of puck-carrying robots was used to derive

$$G(N) \equiv \frac{F(N)}{H(N)}. \tag{2}$$

$F(N)$ and $H(N)$ are analogous functions to $f(N)$ and $h(N)$ from (1), but in this case refer to the interaction of the group of agents, rather than individual agents. The expression above must be a monotonically decreasing function of cluster size in order for larger clusters in a multicluster system to increase in size, since the design of the swarm system will tend to create lower equilibrium densities of puck carrying agents around larger clusters. In situations in which there are multiple clusters, this single condition is strong enough to show that the smallest cluster is absorbed by all other clusters.

These basic results are true whether or not the agents use some "intelligent" mechanism of control. Moreover, the results do not depend on the pucks being pucks, and may be applied to other systems with little alteration as shown in the next section.

3 Robot clustering

Puck clustering is a long process which, because of the time required, is not a feasible choice for construction. It is very tedious to move puck clusters from one place to another. If clusters are formed by robots and not pucks, then robots can flock to move clusters to desired locations. Moving robot clusters is orders of magnitude faster than moving puck clusters.

Robot clustering, though, has potential difficulties which include the following. If all the robots were to start moving at once, there would be no global coordination, and it would be easy for the robots to separate themselves into clusters, but difficult to maintain one cluster. Whatever mechanism is chosen, there must be global information to maintain coordination. This necessary global information can be as simple as regional anchoring.

3.1 Properties of Robot Clustering Systems

We examine a system of autonomous robots with no leader or structure. In our simulations, a moving robot perceives a stationary robot, or a group of stationary robots, as a cluster. If we apply the puck clustering theory, we must produce a monotonically decreasing g function as we did for the generation of puck clusters. This g function is the ratio between the probability of a robot to start moving, or removing itself from a cluster (f), and the probability of a robot to stop, or become a member of a cluster (h). We can easily employ a version of f in which, f decreases as the number of elements in the cluster increases, and h is defined to be 1. Therefore, g will decrease with the formation of a cluster in relation to the cluster size.

To accomplish this, the behavior of the robots can be discussed as follows:

1. Robots employ a memory element that varies between 0 and 1.
2. Robots encountering other robots stop moving upon physical contact with other non-moving robots. This does not reset the memory element. (This is implementation of $h \equiv 1$.)
3. While stopped, robots rotate the sensor direction.
4. While stopped, robots continually decide whether or not to start moving again. If a robot does not see another robot, its memory element is decremented every iteration to decide whether it will remove itself from the cluster it is a member of, or not. This is shown mathematically as follows, where m_1 is the original memory element and m_2 is the new, updated memory element.
$$m_2 = m_1 - (0.0003 \times m_1) \qquad (3)$$
The constant 0.0003 was arrived at through trial and error during a search for optimal parameters, based on the desire to balance the decay of the memory element by a process of resetting it. If $m_2 < 0$ then m_2 is set to zero, causing the robot to remove itself from the cluster. A random number is generated, and if it is higher than the memory element, then the robot moves.
5. If the robot sees other robots, then the memory element is reset to 1.

The probability of a robot removing itself from a cluster is modeled by

$$f(N) = \prod_{i=1}^{n(N)} (i - t_i). \qquad (4)$$

t_i is the threshold, which is a function of the time that the robot does not see a cluster. $n(N)$ is a function of the perceived size of the cluster N and is equal to the rotation steps that the robot makes whilst not seeing the cluster that it is a member of. A robot is unable to see the cluster it is a member of when the orientation of the sensor θ is out away from the cluster. The probability of each stationary robot becoming active increases in relation to the time it does not see a cluster, but decreases in relation to the perceived size of the cluster of which it is a member.

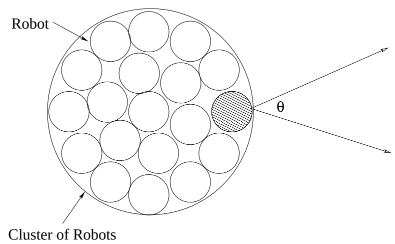

Figure 3.1: This figure shows how a member of a cluster may be unable to see the cluster of which it is a part.

n(N) can also be expressed by the relationship between a robot's perceived angle (θ), which is the angle at which a robot sees something, and its turn ($d\phi$), which is the amount the robot turns before re-evaluating its decision to remain as is, or depart from the cluster.

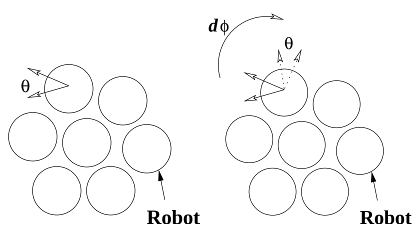

Figure 3.2: One turn of the robot's perceived angle direction is represented by $d\phi$.

$$n(N) = \frac{\Theta(N)}{d\phi} \tag{5}$$

where

$$\Theta(N) = f(r, N) \tag{6}$$

and $\Theta(N)$ varies with the curvature of the cluster. The size of a cluster is approximated by using the robot's angle.

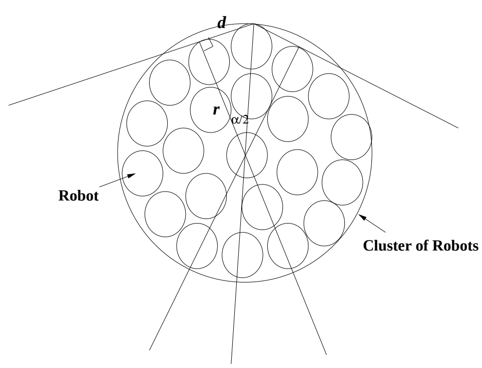

Figure 3.3: This shows how a robot determines the size of the cluster of which it is a member.

The radius of the cluster (r) is *dtan(*α*/2)*, where d is the distance from the viewing robot to the viewed robot, and α is the robot's solid angle. This approximation becomes more and more accurate with the infinite size limit.

3.2 Characteristics

It was shown in [8] that pairwise interaction is not enough to maintain global coordination. It may be sufficient for a few agents that are within a certain proximity, but it does not guarantee coherence throughout an entire group of agents. Global information or a coordination mechanism may provably hold the members of the system together. Without it, there cannot be global coordination. In robot clustering, the global information is the positions of each robot. In robot clustering systems, if all the robots were to move at once, clustering would not result.

A current disadvantage of robot clustering is that multiclustering does not seem possible, while it is possible with puck clustering. This is not possible due to probable coalescence.

4 Simulations

In our experiments we utilize a two-dimensional simulation that models many properties of the real world. These properties include the sensory capabilities, size, and movement of robots. We apply our formalism to the robots in the simulation to thoroughly examine the theory in a more realistic environment. Both the arena and the robots have definite sizes. Each robot is capable of performing two basic actions: deciding to move, and deciding to stop. In this simulation, the robots are of a circular shape and are able to see and identify other robots within a given distance within its viewing angle[1].

[1] In reality, these sensors would be the same as those used by robots involved in puck clustering. These sensors include video cameras and IR sensors.

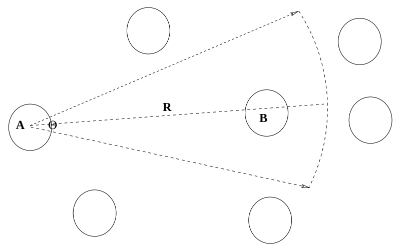

Figure 4.1: This shows a stationary robot deciding whether to start moving or not. θ is the viewing angle and R is the viewing distance.

Simulations take place in an unbounded arena with only robots randomly placed in it. Each robot has a low probability of becoming active, and so a small number of individual robots eventually begin to move. As illustrated in Figure 4.1, a robot is limited to how much it can see. At the moment, robot A is only able to see robot B. Robot A then determines whether robot B is moving or is stationary[2]. If robot B is moving, or no robot is seen at all, then the robot A begins to spiral out from its position, until it finds other robots. If robot B is stationary, robot A will move straight toward robot B, and become stationary if robot A runs into robot B. This implements the $h \equiv 1$ behavior of section 3.

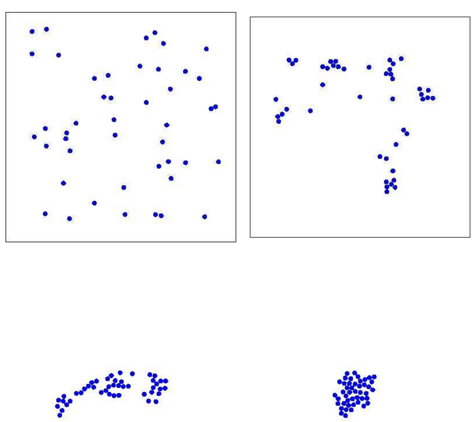

Figure 4.2: Above is a series of screen-shots that displays the clustering process.

A typical run is illustrated in Figure 4.2. Robot clustering systems may be characterized by the number of active robots in the system, the average variance of the robots from a determined center. The number of active robots and the average variance of the robots both decrease as cluster is formed, mirroring the analogous result from puck clustering.

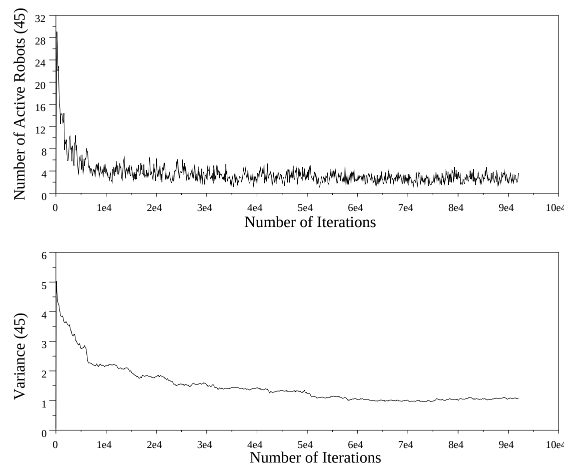

Figure 4.3: The top graph shows how the number of active robots in the system decreases as cluster as cluster is formed. The bottom figure shows the average variance of the robots from a determined center.

As Figure 4.3 illustrates, there are always active robots, even if clustering has been completed because robots around the perimeter of the cluster are always becoming active with some small probabilty. The average variance also declines quickly as the robots become clustered in one location. This is a good indicator of the completion of clustering.

5 Discussion and conclusion

One of the long term goals of swarm-based systems is the use of swarms as construction enablers. As a result, a number of researchers have examined construction mechanisms. Puck clustering systems have been examined as potential methods for initial clustering systems, however, puck clustering systems are impractical as solutions to construction problems because cluster movement takes too long to complete. In order to retain the properties of cluster-based systems we have explored robot clustering systems. Like their puck-based counterparts, our system requires little local information. In this system, the robots are able to form clusters autonomously without the need for global coordination. Thus, it is possible to

[2] In real life situations, robot A would have some sort of motion detecting algorithm, or robot B would have some way of signaling its state by use of a flag or a light.

use robot clustering as a precursor to the movement of these clusters into specific predetermined relative locations.

We explore our robot clustering system using two aspects of the system: the number of active robots and the average variance of the robots from a determined center. The number of active robots decreases as a single cluster is formed, mirroring the analogous result from puck clustering. There is a sharp decline in the average variance of the robots, indicating a rapid coalescence of the robot swarm to form the final cluster. These data illustrate the formation of a single cluster in the system.

What we have demonstrated in this study is that robot clustering can be achieved using the same theoretical considerations used in puck clustering. It is therefore unnecessary to develop new theory specifically for use on robot clustering systems. The local behaviors can simply be developed using the existing theory as a guide. This flexibility demonstrates that clustering systems may potentially be a much larger class of systems than previously thought, as they seem to include both puck clustering systems and robot clustering systems.

Unlike the puck clustering studies, in which all the robots were active at all times, this particular study has made use of a behavior which requires the robots to be inactive much of the time. This behavior is very important, as it preserves the cluster's position, and thereby allows the robots to cluster in a static position. This makes it possible for the different clusters to interact with one another, while keeping the swarm intact.

We intend to continue this work by extending it to include robot segregation studies and multiclustering studies. Later work will extend this to the full cluster positioning studies. Moreover, it is interesting to ask questions about which other systems the same theoretical results of puck clustering systems and robot clustering systems can be applied to. These studies are forthcoming.

6 Acknowledgments

M. Kim is supported by a grant from the Unversity of Southern California Integrated Media Systems Center under an NSF Engineering Research Center grant.

References

[1] J. Deneuboug, S. Godd, N. Franks, A. Sendova-Franks, C. Detran, L. Chretien. *The dynamics of collective sorting. Robot-like ants and ant-like robots.* From Animals to Animats. J.A. Meyer and S.W. Wilson, eds. Cambridge, MA: MIT Press/Bradford Books, 1991.

[2] I. Karsai. *Decentralized control of construction behavior in paper wasps: an overview of the stimergy approach.* Artificial Life, 5(2), pp 117-136, April 1999

[3] S. Kazadi. *Mathematical dynamics of puck clustering systems.* Proceedings of the SCI2003 Conference, Special Session on Swarm Engineering, Orlando, Florida, USA, 2003.

[4] S. Kazadi, A. Abdul-Khaliq, R. Goodman. *On the convergence of puck clustering systems.* Robotics and Autonomous Systems, 38 (2), 93-117 ,2002.

[5] I. D. Kelly, D. A. Keating. *Flocking by the fusion of sonar and active infrared sensors on physical autonomous mobile robots.* The Third International Conference on Mechatronics and Machine Vision in Practice, Guimaraes, Portugal, pp.1/1-1/4, 1996.

[6] C. Melhuish. *Exploiting domain physics: using stigmergy to control cluster building with real robots.* European Conference on advances in Artificial Life, pp. 585-595, 1999.

[7] C. Melhuish, J. Welsby, C. Edwards. *Using templates for defensive wall building with autonomous mobile ant-like robots.* Technical Report Series, Manchester University, U.K, 1999

[8] A. D. Nguyen, Q. P. Ha, S. Huang, H. Trinh. *Observer-based decentralized approach to robotic formation control.* Australasian Conference on Robotics and Automation, 2004.

[9] C. W. Reynolds. *Flocks, heards, and schools: a distributed behavioral model.* Computer Graphics, 21(4), pp.25-34, 1987.