Building Blocks for Multi-Robot Construction

Justin Werfel

Massachusetts Institute of Technology, Cambridge, MA, USA jkwerfel@mit.edu

Summary. One notable capability of social insect colonies that has traditionally inspired distributed robot systems is their construction activity. In this paper, I describe a system of simple, identical, autonomous robots able to build two-dimensional structures of arbitrary design by rearranging blocks of building material into desired shapes. Structure design is specified compactly as a high-level geometric program; robots translate this program into physical form via their fixed behavioral programming. Robots are interchangeable both within and between construction projects, and need not be individually reprogrammed between dissimilar projects. Such a construction team could be used as the first stage in a system for remote building of structures, laying out the floor plan that a more sophisticated system could extend upwards.

1 Introduction

A primary inspiration for distributed multi-robot systems is the set of orders of social insects, notably ants, termites, and bees, whose swarms or colonies accomplish many complex high-level tasks through the collective actions of lower-level agents. One of the most characteristic of these tasks is the robust construction of large-scale, complicated structures, despite the insects' own small size and limited complexity. A corresponding research pursuit is the engineering of multi-robot systems that build specific desired structures, while retaining advantageous features of the insect systems that inspire them (flexibility, robustness, etc.). The possible uses for structure-building teams of robots are many and far-ranging, from automating the production of low-cost housing to allowing construction and related activities in settings where human presence is dangerous or problematic. This latter class in turn ranges from uses in disaster areas, to the construction of first-stage bases of operations to await the arrival of pioneers in, for example, underwater or extraterrestrial environments.

In this work, I describe the design and simulation of a system of simple, identical, autonomous robots able to build structures in the shape of arbitrary non-crossing curves in the horizontal plane, by rearranging blocks of building material into desired shapes on a grid. The shape is specified compactly by a high-level geometric program stored in a separate beacon, which serves as the reference point around which

all robot activity occurs. Robots receive the program for the structure shape from the beacon at short range during the course of the construction project, and translate it into the appropriate arrangement of blocks via their behavioral programming. Thus the same robots can be used in any construction project without needing to be reprogrammed. The intended method of operation is to scatter a handful of generic robots in the vicinity of sufficient building material, place a beacon preprogrammed with the desired structure design, and let construction proceed without further intervention. This system is an example of those for which the goal is to robustly generate prespecified global behavior from local interactions among myriad unreliable components [1].

1.1 Previous work

Most previous work on autonomous construction teams has focused on other aspects of the problem. In [20], robots build a linear wall out of blocks held together by Velcro of alternating polarity. Their multi-robot simulations focus on the benefit of explicit communication, showing that when robots broadcast one bit indicating the polarity of the last block placed, the number of attempts to place blocks of inappropriate polarity is reduced. However, they do not address the issue of specifying more complex structures, nor consider more extensive communication in their building strategies. [10] describes a system of physical robots with force sensors only, that work without explicit cooperation or communication to clear an area of material, by pushing it to the edges of a gradually expanding clearing. [8, 9] describe minimalist approaches to sorting and construction, which have the advantage of simplicity but are typically slow, probabilistic (relying on the correction of frequent errors), and relatively inflexible in the range of structures they can generalize to building. [5] outlines a project whose goal is robots that build 3-D arches and walls at human scale; its robots are intended to work independently rather than collaboratively, and its primary concern is with mechanical engineering considerations, with no reference to the question of controlling high-level building design. Its approach is that of [3,4], whose simulations consider the inverse problem of studying the kinds of structures that result from different simple rules for agent behavior, but do not address the issue of generating prespecified high-level structures.

A related topic is the regulation of formations of agents. Such approaches can be applied directly to construction if building blocks themselves are mobile robots. Some approaches to formation control require continuous global knowledge about all agents, and/or user intervention [2, 6, 15]; others can generate crystalline formations, but do not lend themselves to the design of high-level forms [14]; reconfiguration algorithms for modular robots create two- or three-dimensional forms out of agents which are not arbitrarily mobile, but remain always in contact with one another [18, 19].

In contrast to the preceding, this work focuses on a system of mobile robots with local knowledge and local interagent communication. These arrange passive building materials in the horizontal plane into arbitrary non-crossing curves, which can be easily prespecified by the user. Mobility and structural requirements are separated,

allowing the design of each class of elements to be specialized, the more sophisticated elements (robots) to be reused for multiple projects, and the passive elements (building materials, which after installation need never move again) to be of minimal manufacturing difficulty and cost.

2 Component capabilities

Objects in the world of this system are mobile robots, a fixed beacon, and passive, movable blocks, all of which are initially scattered at random over the workspace.

Robots are assumed to possess the following abilities: move in any direction unless obstructed, and detect if intended movement in a direction is impeded due to some obstacle; pick up, carry, and put down blocks (carrying a block may increase the 'footprint' a robot occupies, which in turn may affect how it must plan trajectories in some cases); recognize blocks and other robots when close to them (in the simulations described here, 'close' was four body-lengths); communicate with other robots within that distance, exchanging information and commands; and detect and evaluate the direction and strength of a signal emitted by the beacon. With the exception of that latter long-distance signal, robots are restricted to local information about their immediate surroundings only.

The beacon broadcasts a long-range, low-bandwidth signal which can be detected by robots from anywhere in the workspace. It cannot obtain long-distance information about the status of robots or the progress of the task; thus the primary utility (and motivation) of the broadcast is as a reference to orient to. The beacon can communicate with robots that are near enough, just as they communicate with each other.

Blocks in this work are taken to be identical, so that robots need not be confronted with the additional problem of determining how to manipulate heterogenous blocks in varying circumstances.

3 Methods

The simulation was written in Swarm, a free objective-C-based system available at `http://www.swarm.org`. Many details of the model were simplified away for this preliminary study. Most immediately, the simulation took place on a two-dimensional cellular grid; thus robots and blocks each occupied exactly one cell, robots were restricted to move in the four cardinal directions, and issues of fine position adjustment were sidestepped.

Before deployment of the system, the beacon is programmed with the design for the desired final structure. This program takes the form of a list of corners; each specifies its distance from the beacon, the angle (positive or negative) to the next corner, and whether a wall between the two is to be straight, curved (perpendicular to the signal gradient everywhere), or absent altogether (Fig. 1A). Such a list completely specifies the structure's geometry, though not its orientation; if robots or the beacon are equipped with a compass, additionally establishing a desired building orientation is trivial.

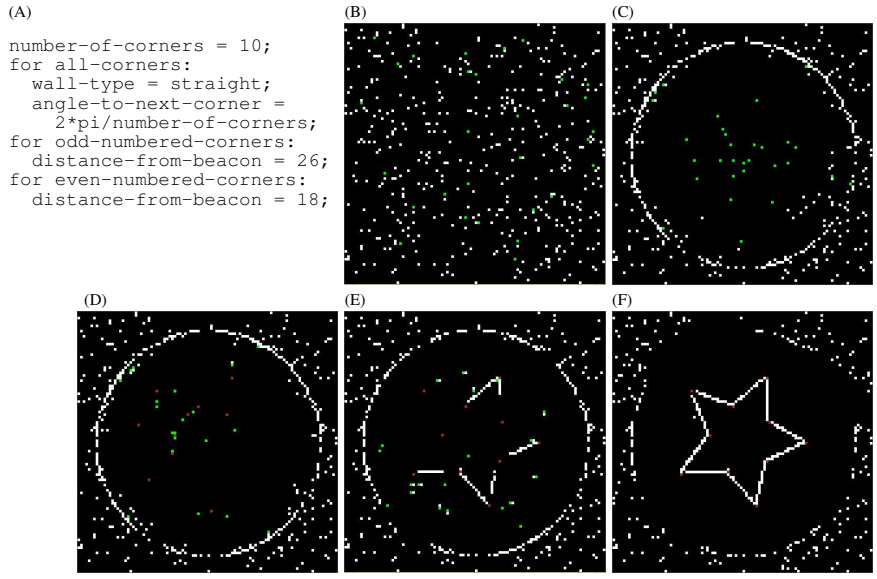

Fig. 1. Example of a structure program and snapshots of several steps in the construction of the associated structure. **A**: pseudocode program for a star-shaped building. **B-F**: stages in the construction of that building. **B**: initial state, with blocks (white) and robots (green) scattered randomly, and beacon (not shown) at center. **C**: First the robots clear a space to work. **D**: Some robots take on the role of embodying corners (brown) and begin to localize themselves according to the building program. **E**: When corners are placed, the remaining robots begin to build walls between them. **F**: Final structure (only blocks and corners shown).

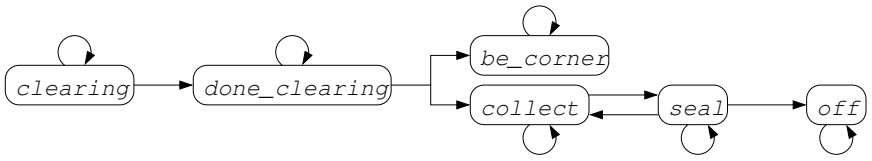

Fig. 2. FSM for behavioral mode. Robots start in the *clearing* state.

3.1 Robot behavior

The algorithm the robots follow can be described as follows (also see Fig. 1). A finite state machine (FSM) specifies each robot's high-level behavioral mode (Fig. 2). All robots start in *clearing* mode: they follow the signal to the beacon (noting its position), then spiral outwards. If a robot encounters a block at any point along the way, it picks it up and carries it directly outward until the signal strength from the beacon falls below some predefined threshold; the robot then returns to the beacon and repeats the process. If, while spiraling out, it reaches that signal threshold without encountering any blocks, or if it encounters a robot in any mode other than *clearing*,

the robot enters *done_clearing* mode: it spirals back inwards, in the opposite direction to increase the number of *clearing* robots encountered, to bring the entire robot population onward to the same mode and avoid the problem of having some robots working on building the structure while others work just as hard to clear it away. Upon reaching the beacon, the *done_clearing* robot receives a new assignment.

The beacon contains the program for a C-corner structure, as described above. The first C robots that come to it in *done_clearing* mode are assigned to act as successive numbered corners, and enter *be_corner* mode. The first of these moves outward from the beacon to the appropriate radius, using odometry and beacon signal strength to estimate distance, and immobilizes itself there. Each succeeding corner is specified in relation to the previous one; the corresponding robot circles at the radius of its predecessor, until it finds that previous robot fixed in its final location, or encounters another robot that knows that location; it then calculates its own destination location on that basis[1], and goes and immobilizes itself there.

A robot after the first C that reaches the beacon receives the building design, randomly chooses a pair of successive corners between which a wall is to be built, and enters *collect* mode: First it must know the locations of its selected corners, which it finds either by seeking them out itself or by being told their locations by robots it encounters which already know. During this stage it circles in the opposite direction to other robots, again to increase the rate of unique encounters. Next, it goes out beyond the outskirts of the cleared area to find a block, takes it to the first of its two corners, and follows the line between the two (straight or curved as appropriate) until it finds a valid unoccupied position to place its block. It does this by calculating the location of the nearest point to itself on the desired wall, i.e., the perpendicular to the line or arc connecting the two corners, based on the known positions of those corners and the type of wall desired. It then moves within sensor range and looks to see if that cell is occupied.[2] If not, it goes on to try to place the block there; otherwise, it moves along the direction of the desired wall, and will check the corresponding new perpendicular location on the next time step.

Robots repeat this process until they reach the second corner without finding a place that needs a block, at which point they enter *seal* mode; they return back along the wall to the first corner, making sure there are no gaps they missed the first time. More elaborate future versions of the system might have robots, for instance,

[1]Note that each robot must by necessity maintain its own private coordinate system. In general, each robot's coordinate system may permissibly differ from those of the others by rotation, translation, and scaling; common reference points can be used, whenever two robots exchange information, to calculate the appropriate linear transformations to convert between the two systems for that interchange. See also the discussion on localization in §4.

[2]In the present instantiation, robots do not distinguish between occupation by carried blocks, placed blocks, or other robots; this may lead to temporary bypassing of locations that would have opened up a few time steps later when the blocking robot moved on, but it also helps avoid traffic jams (the robot in the way may in turn be waiting for the first robot to get out of its own way so it can leave the area), and the gap can be filled in during a later pass by any robot; also, distinguishing between carried and placed blocks would require more sophisticated identification capabilities in a hardware implementation of this system.

spray some sealant over the blocks for airtightness during this stage. If the robot finds a gap, it fills it with a block and returns to *collect* mode; if it reaches the first corner still in *seal* mode, it records that wall as completed, reenters *collect* mode, and chooses another pair of corners to work on the wall between, until in the end it has personally verified that all walls specified by the building program have been completed. At that point, the robot enters *off* mode: it heads away from the beacon to the outskirts of the workspace, to avoid interfering with any other construction that may still be ongoing, and ceases to be active. In more complicated situations, where other construction tasks on other parts of a more complex building still remain, or the structure is subject to damage and requires constant maintenance, etc., the appropriate behavior would be to continue to *collect* rather than turning *off*.

Additions to this basic algorithm handle special situations. If a robot wants to place a block somewhere but is prevented from doing so for too long, it will give up and move on. An robot unable to move at all for too long will send out a signal to all robots within range, on receipt of which robots will shuffle around at random for several time steps, in the hopes of breaking up a traffic jam if that was causing the problem (as can occur when more than a few robots are at work in the same area).

While robots are capable of locating gaps in and adding blocks to a wall from either side, their behavioral algorithm favors construction from the side away from the beacon, and the supply of free blocks is located on the outskirts of the building area. Consequently, in complicated structures with corners at different radii, early completion of the more outlying walls can interfere with subsequent work on the inner ones. This problem is materially avoided by having robots choose first to work on walls adjoining the structure's smallest-radius corners before they move on to those of larger radius. Other exceptions to the basic algorithm above respond to an environment that may change in significant ways between the time when a robot begins an action and the later time when it completes it; for example, a robot heading to claim a block which another picks up before the first reaches it will return to its previous goal and continue to search.

4 Results and discussion

By changing the geometric program stored in the beacon, the system can quickly and easily be made to produce a wide variety of 2-D structures with non-crossing walls. Fig. 3 shows several examples, giving a sample of the system's flexibility and range. The time course of construction with the same building program but different initial conditions was similar across runs with independent random seeds (Fig. 4A).

A distributed system may derive its effectiveness simply from its intrinsic parallelism; or it may take advantage of explicit cooperation between multiple agents. The system described here takes the former approach, for the most part, with robots largely ignoring one another while going about their behaviors. We might then naïvely expect the building task to be completed, for an N-robot system, in $1/N$th the time it would take a single robot. However, this incremental advantage is diminished as the number of robots increases, since any task has a limit to how many

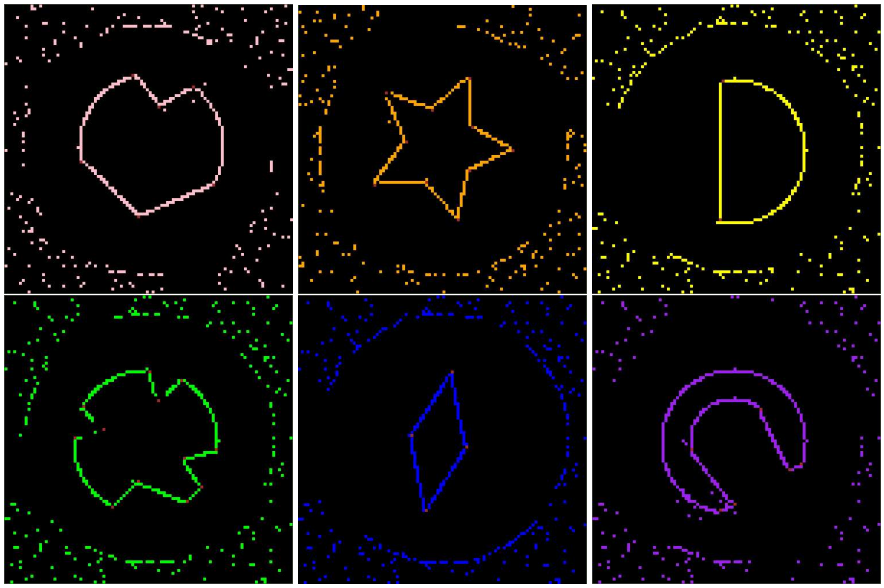

Fig. 3. Different structure programs can direct the system to build a variety of shapes (in this set of examples, the marshmallow shapes of General Mills's Lucky Charms cereal). Only blocks (colored according to marshmallow shape) and robots acting as corners (brown) are shown. All simulations were conducted on a 100×100 lattice with a team of 30 robots.

agents can usefully contribute to its completion at one time, and robots begin to get in one another's way (Fig. 4B, C). Communication between agents, in general, can help reduce such interference [20]. Here, communication was useful for alleviating traffic jams, in that robots unable to move for too long signaled any nearby to shuffle their positions, often breaking impasses; for coordinating the operating mode, to keep robots from working in direct opposition (e.g., one bringing blocks in, another clearing them away); and for finding the locations of corners, which robots obtained more quickly through the team's distributed search than they would have alone.[3]

A further advantage of communication could be used to address an important limitation of this model, the difficulty in real systems of localization. Odometry alone is unreliable, as sensors are noisy, actuators are imperfect, and an isolated robot's estimate of position becomes increasingly unreliable as errors accumulate. Methods have been developed for individual robots for slowing [13] and bounding [17] this drift. What is more, the multi-robot nature of the system can itself be taken advantage of; robots exchange position and orientation estimates whenever they encounter one another, and using the information provided by the other, each can improve its own estimate to obtain a significant decrease in uncertainty [12]. The ubiquitous signal

[3]Functions which, like the latter two examples, are fundamentally global could potentially be further facilitated via appropriate modulation of the beacon broadcast, on the basis of information robots carry to the beacon during the course of construction.

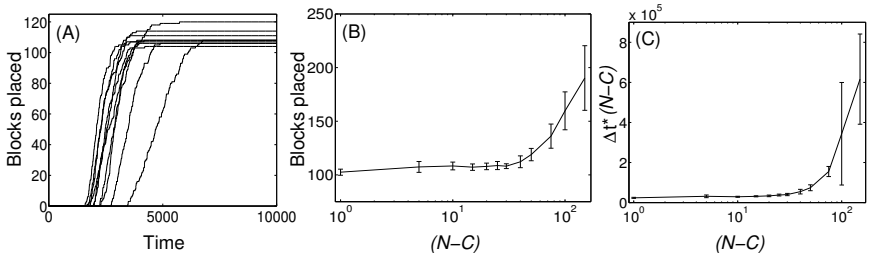

Fig. 4. Aggregated results from many independent runs building the diamond shown in Fig. 3. **A**: For 30 robots, number of blocks placed by robots as part of the structure (not simply dropped at the periphery after being cleared away) as a function of time, for 10 runs with different random seeds. The variation in final numbers is due to occasional remaining gaps and blocks extraneously placed, as are visible in Fig. 3.
B, C: Interference between robots affects number of extraneous blocks placed and proportional time taken to complete the task. With N robots and $C = 4$ corners, $(N - C)$ is the number of robots available to manipulate blocks after the clearing stages. Each data point represents 10 runs. **B**: Total number of blocks placed as part of the structure. **C**: Time between when the first block of the structure was placed and that when 95% of all such blocks had been placed, multiplied by $(N - C)$. For fewer than about 40 robots, interference was small or negligible.

from the beacon will provide another cue that can be used to improve the position estimate; and the beacon itself, and (once in position) the robots that embody corners, represent fixed landmarks that a robot can use to correct its estimate whenever it comes near them, which it will do frequently in the course of construction.

A clear motivation for the use of distributed systems in general is to improve robustness. While this issue has not yet been studied in these simulations, we can discuss how this system would withstand component failure, and how its response could be improved in those cases where the instantiation described here would do poorly.

- Loss of individual unspecialized robots (i.e., those not acting as corners) would have no significant effect; because they are interchangeable and working independently, loss of one or several would slow construction comparatively little.
- The loss of corner-robots would be more problematic, and without some added system response, construction could halt. To deal with the risk of loss of a corner-robot before it had found its final destination, a sufficient approach would be to specify that if a robot circles too many times without finding the corner it seeks, it takes on the task of embodying that corner for itself (first returning to the center to notify the beacon, so that if it had previously been tasked with embodying another corner, that task can be reassigned). If, on the way to embodying a corner, a robot encounters another one that has already planted itself at the appropriate location as that corner (or if it learns of such a robot from a third party), it reverts to acting as an unspecialized robot. As for corner-robots that fail after positioning themselves, these should not pose a significant threat to the task, since all corner-

robots must do is act as a landmark; if another robot comes looking for that corner and finds a robot which is in the right place but not communicating, the newcomer need only rely on its own sensors rather than the corner-robot's account of its location, and the only cost is a possible increase in positional uncertainty in the vicinity of that corner.

- Loss of the beacon at first seems more fatal still; without its signal as a constant reference, robots will have to rely on their position estimates alone in planning trajectories, and the final construction will be more irregular at best, incomplete at worst. Moreover, if the beacon is lost early enough, corners may go unassigned, the building program may never be communicated to the robots, and robots may have no common basis even for a position estimate. A more robust approach, then, would be to build the potential to act as a beacon into each robot. Rather than having the robots receive the building program in the course of construction, they could receive it before being deployed, when all are close together, via a general broadcast. Then, at the start of the construction process before any beacon has yet existed, each robot can choose to become a beacon at random with low probability per unit time; as soon as one does, the others orient to it and begin the construction process as before. If the beacon later fails, the loss of the long-range signal leads the other robots to put their current tasks aside and head for where it had been; whichever first gets close enough locates the previous beacon, takes its place, and adopts the beacon's role from that point on while the other robots return to work.

In this report, I have described a model system which in simulation allows highly flexible construction of 2-D structures, specified in a simple high-level geometric language, through the distributed actions of many identical, autonomous robots. A straightforward extension of this approach could achieve structures of multiple closed rooms or where corners can be endpoints of more than two walls; fully three-dimensional structures, a greater challenge, are its ultimate aim. The high-level features of the system described here may be useful to consider in design of hardware implementations of robots intended for autonomous construction projects [5], as well as studies of tasks requiring explicit cooperation between multiple robots [7], heterogenous teams of robots [7, 11], and other related work and its future development.

Acknowledgements

This work was supported by NIH training grant GM07484, with additional support from a Packard Foundation Fellowship (to H.S. Seung). I would like to thank Radhika Nagpal for useful discussions and comments on the manuscript, as well as Gerald Sussman and Hal Abelson.

References

1. Abelson, H., et al. (2001). Amorphous computing. *Communications of the ACM* **43**(5): 74–82.

2. Bahceci, E., Soysal, O. & Sahin, E. (2003). *A Review: Pattern Formation and Adaptation in Multi-Robot Systems* {Technical Report CMU-RI-TR-03-43, Carnegie Mellon Univ.} Pittsburgh, PA, USA.
3. Bonabeau, E., Dorigo, M. & Théraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press Inc.
4. Bonabeau, E., Theraulaz, G., Deneuborg, J.-L., Franks, N., Rafelsberger, O., Joly, J.-L. & Blanco, S. (1998). A model for the emergence of pillars, walls and royal chambers in termite nests. *Phil. Trans. R. Soc. Lond. B* **353**: 1561–1576.
5. Bowyer, A. (2000). *Automated Construction using Co-operating Biomimetic Robots* {Technical Report, University of Bath Department of Mechanical Engineering}. Bath, UK.
6. Fredslund, J. & Matarić, M. (2001). *A General, Local Algorithm for Robot Formations* {IRIS Technical Report IRIS-01-396}. Los Angeles, CA, USA.
7. Gerkey, B. & Matarić, M. (2002). Pusher-watcher: an approach to fault-tolerant tightly-coupled robot coordination. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, D.C., USA: 464–469.
8. Melhuish, C., Holland, O. & Hoddell, S. (1998). Collective sorting and segregation in robots with minimal sensing. In *5th Conference on Simulation of Adaptive Behaviour*, Zurich, Switzerland.
9. Melhuish, C., Welsby, J. & Edwards, C. (1999). Using templates for defensive wall building with autonomous mobile ant-like robots. In *Proc. Towards Intelligent Autonomous Mobile Robots 99*, Manchester, UK.
10. Parker, C., Zhang, H. & Kube, R. (2003). Blind bulldozing: multiple robot nest construction. In *Proc. IROS 2003*, Las Vegas, USA.
11. Parker, L. (2003). The effect of heterogeneity in teams of 100+ mobile robots. In *Multi-Robot Systems Volume II: From Swarms to Intelligent Automata*, Kluwer: 205–215.
12. Roumeliotis, S. & Bekey, G. (2000). Distributed multi-robot localization. In *Proc. 5th International Symposium on Distributed Autonomous Robotic Systems (DARS 2000)*, Knoxville, TN, USA: 179–188.
13. Roumeliotis, S., Sukhatme, G. & Bekey, G. (1999). Smoother based 3-D attitude estimation for mobile robot localization. In *Proc. 1999 IEEE Int. Conf. in Robotics and Automation*, Detroit, MI, USA: 1979–1986.
14. Spears, W. & Gordon, D. (1999). Using artificial physics to control agents. *IEEE Int. Conf. on Information, Intelligence, and Systems*, Bethesda, MD, USA.
15. Sugihara, K. & Suzuki, I. (1996). Distributed algorithms for formation of geometric patterns with many mobile robots. *J. Robotic Systems* **13**(3): 127–139.
16. Théraulaz, G. & Bonabeau, E. (1995). Modelling the collective building of complex architectures in social insects with lattice swarms. *J. Theor. Biologie* **177**: 381–400.
17. Thrun, S. (1999). Learning maps for indoor mobile robot navigation. *Artificial Intelligence* **1**: 21–71.
18. Vassilvitskii, S., Yim, M. & Suh, J. (2002). A complete, local and parallel reconfiguration algorithm for cube style modular robots. In *Proc. 2002 IEEE Int. Conf. on Robotics and Automation*, Washington, DC, USA: 117–122.
19. Vona, M. & Rus, D. (2001). Crystalline robots: self-reconfiguration with compressible unit modules. *Autonomous Robots* **10**(1): 107–124.
20. Wawerla, J., Sukhatme, G. & Matarić, M. (2002). Collective construction with multiple robots. In *Proc. 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne, Switzerland.
21. Werger, B. & Matarić, M. (1999). *Exploiting Embodiment in Multi-Robot Teams* {IRIS Technical Report IRIS-99-378}. Los Angeles, CA, USA.