Robot Formations Using Only Local Sensing and Control

Jakob Fredslund*
Computer Science Dept.
Aarhus University
Aabogade 34
8200 Aarhus N
Denmark

Maja J Matarić†
Interaction Lab
Computer Science Dept.
University of Southern California
Los Angeles, CA 90089
USA

Abstract We study the problem of achieving global behavior in a group of robots using only local sensing and interaction, in the context of *formations*, where the goal is to have N mobile robots establish and maintain some predetermined geometric shape. We have devised a simple, general, robust, localized, behavior-based algorithm that solves the problem for N robots each equipped with sonar, laser, camera, and a radio link for communicating with other robots. The method uses the idea of keeping a single *friend* at a desired angle (by panning the camera and keeping the friend centered in the image), and only communicating heartbeat messages. We also developed a general analytical method for evaluating formations and applied it to our algorithm. We validate our algorithm both in simulation and with physical robots.

1 Introduction

A university football marching band displays coordination in addition to musical skill. Each member has very limited knowledge of the others' positions, so the conductor, who has complete knowledge, overlooking the group, plays a major role. Imagine the conductor being one of the band members: down on the field, with no more knowledge or sensing capabilities than the next person — a radically different coordination strategy would be needed.

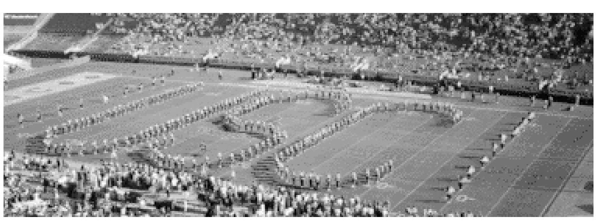

Figure 1: USC's football marching band.

In this work we have studied the problem of achieving global behavior in a group of robots using only local sensing, and we have considered *formations* as an instance of that general problem. Here the goal is to have N mobile robots establish some predetermined geometric shape, then maintain or re-form that shape, or change to another shape, while negotiating obstacles and experiencing occasional fallouts of group members. We have devised a simple, general, robust, decentralized, behavior-based algorithm that solves the problem for N robots, each equipped with sonar, laser, camera, and a radio link for communicating with other robots. We also developed a general set of quantitative criteria for evaluating formations and applied it to our algorithm. We validate our algorithm both in simulation and with physical robots.

*Currently visiting Maja Matarić's Interaction Lab

†This work was funded partly by the Office of Naval Research (Grants N000140110354 and N000140010140), and DARPA MARS Program (Grant DABT63-99-1-0015).

2 Related Work

A variety of approaches have been proposed to create global behavior in a group of mobile robots. [8] showed how a set of simple behaviors (avoidance, aggregation, and dispersion), based on local sensing only, can be combined so that a global flocking behavior emerges. In [11], a robot soccer playing team is described that has a minimalist behavior-based control system with only a few basic behaviors. From their interaction, two different group formations emerge, enabling seemingly 'willed' offensive and defensive team play. In [7], as little as two basic, local behaviors (avoidance and goal seeking) were proven by experiments with five physical robots to be enough to result in a successful collaborative box-pushing behavior. [12] presents work where a group of robots, equipped with only touch sensors, are able to form a physical chain reaching from a nest to a source of food. Minimalist communication through the chain is achieved by robots tapping each other's touch sensors. The global chain emerges from a set of basic behaviors that use only local sensing.

With formations, however, a more rigid and reliable structure is needed from the group. Each robot has to determine its spot relative to its peers. In [1], three ways of doing this are identified: 1) *unit-center-referenced*, where the robot determines its position relative to the centroid of all robots; 2) *leader-referenced*, where the robot uses the position of a pre-determined leader; and 3) *neighbor-referenced*, where the robot positions itself relative to a neighbor; however only the first two seem to be tested in the paper. Each robot determines other robots' positions by dead reckoning, GPS, or by direct perception, and its own coordinates in the global coordinate system are broadcast to all robots. Experiments were done with both simulated and real robots. The high reliance upon a centralized world view and the need to transmit coordinates between robots might have a negative impact on performance, as the paper states.

In [2], all robots have a predetermined set of 'attachment sites' spread uniformly around the body, and the formation emerges as the group 'snaps' into shape with robots being 'pulled' towards the nearest attachment site. Depending on the angular offset of the attachment sites, different formations are possible. The approach is validated in simulation. Since there is no one 'right' spot for each robot due to the symmetrical nature of the attachment sites, several configurations with the same attachment sites are possible, while only a specific one may be desired. Other researchers have studied formations in simulation using more theoretical approaches enabling formal performance analysis, e.g., [3, 4].

There is thus a spectrum of strategies, ranging from simple, behavior-based, purely local ones out of which global formations emerge, to more involved ones relying to varying extent on global knowledge, typically a global coordinate system or knowledge of other robots' positions and headings. The former category is characterized by minimalism and robustness but a lack of any guarantees that the desired formation will actually emerge; the latter category by reliability and efficiency but also a need for global knowledge and computation. In [9], Parker defines what 'global knowledge' could mean: knowledge of global goals (Type 1), and/or of the actions and intentions of other robots (Type 2). Within this framework, a robot knowing what formation and with how many robots it is supposed to participate in would have Type 1 global knowledge, whereas its knowing the globally required formation heading or whether another robot is about to evade an obstacle would be global knowledge of Type 2. [9] illustrates how the addition of global knowledge can improve system performance through formation simulations with four robots.

3 Approach

Bearing the spectrum of movement coordination strategies we discussed above in mind, we have in our approach sought simplicity yet reliability through local sensing and minimal communication. Generality is also a primary goal; traditionally, the four formations studied are diamond, column, line (abreast), and wedge, but we have aimed for an algorithm applicable to any geometric shape.

The basics of our algorithm are as follows. Each robot has a unique ID number. One robot is the *conductor*; any robot can take on this role. The conductor defines the heading for the whole formation. All robots regularly broadcast heartbeat messages with their ID. The conductor broadcasts the number of the current formation along with its ID. With the exception of the conductor, every robot follows one neighboring robot called its *friend*, keeping a certain distance and bearing to it. It does so by panning its camera some appropriate number of degrees pertaining to the current formation; thus, maintaining the right position is simply keeping the friend in the center of the image, for all formations.

Each robot can only have one follower, except for the conductor who can have either one or two. The conductor, who is in front of the formation, is thus followed by either one or two cadres of robots. In the latter case, the two cadres are of the same length. The whole structure is called the *chain of friendships*: it either has the conductor at one end, or in the middle (Figure 2). The ordering of the robots in the chain is by ID. Formations with the conductor in the center are called *centered* formations. Any connected shape that can be obtained by folding this chain of friendships, so that no robots have to look behind them for their friends, is captured by our algorithm. Thus, shapes like J, U, and ∼, with upwards heading, are not allowed.

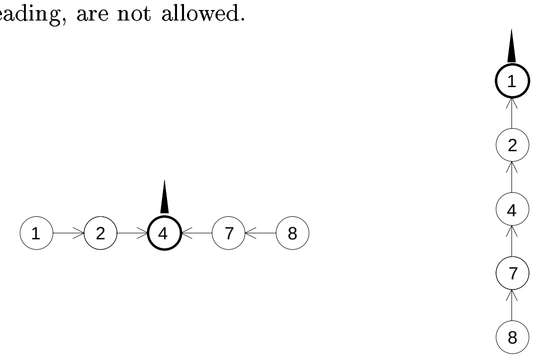

Figure 2: The chain of friendships; $i \to j$ means i looks for j, a black triangle indicates the formation heading. In the centered formation on the left (line), the robot with the middle ID is the conductor, 4. For the non-centered formation on the right (column), the robot with the lowest ID is the conductor, 1.

Thus, each robot senses the relative distance and bearing to its friend, and by radio communication it knows the total number of robots currently participating, and their IDs. It does *not* know the position of any other robot, and it does *not* know the heading of the conductor. There is no shared coordinate system.

This approach offers several nice implications. First, once the conductor starts moving, the only way for a robot to keep a stable position relative to its friend is by finding its heading. In this way, the conductor 'drags' its followers and, by transitivity, the whole formation, into place just by going its own way. No global heading needs to be agreed upon. Since any robot can be the conductor, what seems a centralized element really is not: if the conductor fails, another robot takes over the role. Second, since the algorithm is basically 'keep your friend in the center', a switch between centered formations is easily done by gradually panning the camera into the appropriate angle (*camangle*); the change in position results automatically.[1] Third, there is no global coordinate system and hence no communication of coordinates.

Each robot R has a behavior-based controller consisting of three concurrent behaviors and a module holding state data (Figure 3). Each is described in turn.

The state module *WhatDoIKnow*: Here resides all state information: R's own ID, the total number of robots N, the table of IDs, *lessThanMe* (see below), the current formation f, and *camangle*. The behaviors manipulate and make use of this information as follows.

The *channelNListener* behavior receives the heartbeat messages from the other robots and maintains N and the table of IDs in *WhatDoIKnow*. This information is used to calculate *lessThanMe*, the number of live robots with IDs lower than R's own ID (needed since IDs need not be consecutive). If N changes, R might be promoted to be the conductor. This happens if its *lessThanMe* $= \lfloor N/2 \rfloor$, in which case R is the middle robot.[2] Conversely, R could also be demoted if its

[1] Switches to and from the column, being non-centered, are special cases: they involve switching the conductor between the leftmost and the middle robot, so the robots between the two must find a new friend on the opposite side in the chain of friendships.

[2] In case of the non-centered column, the leftmost robot should be the conductor, i.e., the one whose *lessThanMe* is 0.

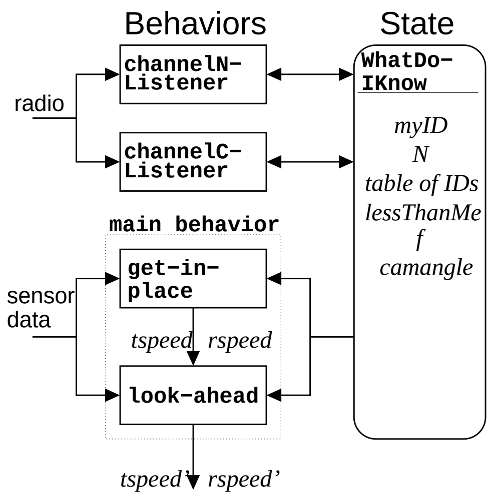

Figure 3: The controller consists of a state module, *WhatDoIKnow*, and three concurrent behaviors. The first two, *channelNListener* and *channelCListener*, receive messages from other robots. The main behavior consists of two serialized sub-behaviors, *get-in-place* and *look-ahead*, that control the robot based on local sensor data.

$lessThanMe \neq \lfloor N/2 \rfloor$. In this case it looks up a friend in its table of live IDs.

The *channelCListener* behavior receives messages from the conductor and updates the formation variable, *f*, in case of a formation switch. If *f* changes, R might have to pan its camera to a new angle. The correct angle, *camangle*, is calculated from a simple geometric relation of *lessThanMe* and *N*.

The main behavior: The robots move by setting two parameters, translational and rotational speed (*tspeed* and *rspeed*). In the main behavior, R cycles through a control loop that reads the sensors, sends out its heartbeat message, sets *tspeed* and *rspeed* to their default values, and then passes them to the *get-in-place* sub-behavior.

The *get-in-place* sub-behavior: If R is the conductor, it pans its camera straight and modifies neither $tspeed$ nor $rspeed$ — unless it has just circumnavigated an obstacle, in which case it modifies $rspeed$ so as to gradually return to the heading it had when it encountered the obstacle. Previous heading is stored in *WhatDoIKnow*.

If R is *not* the conductor, it first looks for its friend, F. Then it attempts to obtain the right distance to F (in our case measured by the laser), and to center F in its image, all by making corrections to $tspeed$ and $rspeed$ proportional to the centering and distance errors. At the same time, once F is close, R starts panning its camera toward the appropriate *camangle*. After all updates have been made to $tspeed$ and $rspeed$, they are sent to the *look-ahead* sub-behavior.

The *look-ahead* sub-behavior: A central element of *look-ahead* is the *aheadbuffer*. From $tspeed$ and $rspeed$, a bounding box for the resulting movement is calculated and a buffer is added: the width of the robots, robot-size, on the sides, and $aheadbuffer$ in the front. *look-ahead* checks if any obstacles are found within this augmented bounding box. If so, a correction is made to $tspeed$ and $rspeed$ proportional to the proximity of the sensed obstacle.

Thus, $aheadbuffer$ induces immediate obstacle avoidance. In addition, if it is set to a high value relative to robot-size, it allows R to look far ahead for obstacles, resulting in an elegant, smooth avoidance behavior (Figure 4). However, R cannot keep $aheadbuffer$ high if it is not yet in its place in the formation, as it may have to get close to other robots. Consequently, $aheadbuffer$ is set *low* (to robot-size) if R has not yet been in place in the formation, R has been out of place for a long time, or R has other robots in front of it when moving in the formation (derived from N, *lessThanMe* and *f*). Otherwise, $aheadbuffer$ is set to a higher value proportional to robot-size and N (a large formation needs comparatively more time to avoid an obstacle). If R is the conductor, $aheadbuffer$ is always set high. As a safety measure, another sensor (in our case sonar) can be used for lowest-level collision avoidance.

Finally, the *look-ahead* behavior sends the revised ($tspeed$, $rspeed$) command to the wheels, and R makes the corresponding movement. Thus, through a series of simple update rules that modify the translational and rotational speed based only on local sensor information, a small set of state variables, and the heartbeat information, each robot eventually falls into place relative to its friend. In turn, a global formation emerges. We have tested the algorithm both in simulation and on real robots and validated it in a battery of experiments, described next.

4 Experimental Evaluation

A standardized, formal definition of the notion of 'being in formation' is beneficial: it allows for direct comparisons between different approaches, and it might give a clearer prediction as to whether an algorithm tested in simulation would work in practice. We therefore propose the following general *formation evaluation criteria*:

Definition1 Given the positions of N mobile robots, an inter-robot distance $d_{desired}$, a desired heading h, and a connected geometric shape $\mathcal{G}$ completely characterizable by a finite set of line segments and the angles between them, the robots are considered to be in formation $\mathcal{G}$ iff:

(1) *uniform dispersion*: $\exists d$, such that $\forall$ pairs of immediate neighbors (R_{i_1}, R_{i_2}) with distance $dist(R_{i_1}, R_{i_2})$, $|d - dist(R_{i_1}, R_{i_2})| < \epsilon_{d_1}$, and $|d - d_{desired}| < \epsilon_{d_1}$,

(2) *shape*: $\exists$ a 'stretching function' f with $f(\mathcal{G}) = \tilde{\mathcal{G}}$, such that $\forall$ angles $\theta \in \mathcal{G}$, $|f(\theta) - \theta| < \epsilon_a$, and such that $\forall$ robots R_i, with distance $dist(R_i, \tilde{\mathcal{G}})$ to $\tilde{\mathcal{G}}$, $dist(R_i, \tilde{\mathcal{G}}) < \epsilon_{d_2}$,

(3) *orientation*: $|f(h) - h| < \epsilon_a$; for small $\epsilon_{d_1}, \epsilon_{d_2}, \epsilon_a > 0$.

Criterion 1 states that the same distance should be kept between all neighboring robots. Criterion 2 states that it should be possible to lay out the desired shape over the position data and adjust the angles so that all robots are close to this shape. No angle in the original shape must be stretched more than ϵ_a to make the data points fit. Allowing the heading of the shape to be one of its defining angles, Criterion 3 states that the stretching from Criterion 2 must not skew the heading more than ϵ_a.

Note that by the term "immediate neighbor", Definition 1 does not demand completeness of formations; this means that 6 robots can actually form an (incomplete) diamond. Note also that the measure is *global*: N robots are not considered to be in line even if the angular offset between neighboring robots is small, if overall they form, say, an arc. In other words, even if the behavior of a robot is locally meaningful, it may not be so when considered globally. Definition 1 is easily extended to shapes that do not consist of line segments, such as circles; we omit these special cases to enhance clarity. We have applied Definition 1 as the formation evaluation measure in all our experiments.

4.1 Simulation Experiments

Our simulation experiments were done using *Player* and *Stage*. *Player* is a server and protocol that connects robots, sensors, and control programs across the network; *Stage* simulates a set of Player devices [6, 10]. Figure 4 shows a sample run; each robot is 25x25 cm. In our implementation, besides a camera, robots utilize lasers (for distance measurements), sonars (for close-up detection), and radios for communication. Our simulation experiments were designed so as to demonstrate the following properties:

1. Stability (robots stay in an established formation)
2. Robustness (formation adapts if N changes)
3. Formation switching
4. Obstacle avoidance (robots re-form after obst.)

Each is addressed below. We used Definition 1 to determine if the robots were in formation, using $\epsilon_{d_1}=\epsilon_{d_2}=10\%$ of the desired inter-robot distance (set to 70 cm), and $\epsilon_a=3.6$ degrees. In all experiments, the robots started in the correct order with respect to the chain of friendships (but not necessarily with their friend in the visual field) and with random heading (except for the conductor, whose heading was correct). We feel that this starting assumption is reasonable, since the problem of aggregating robots into such a formation has already been empirically demonstrated [8]. Furthermore, following ideas from [12], it is possible through only local interaction to have N robots form a chain. Once the chain is established, the distribution of monotonic IDs could follow, and the matching of unique color markers with IDs could be communicated between robots.

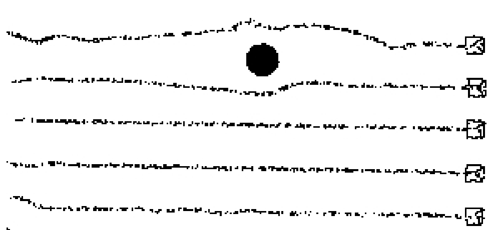

Figure 4: Five robots in a line negotiating an obstacle.

To show **stability**, we ran 10 experiments with 4 robots for each of the basic formations, line, column, diamond, and wedge, where the robots had to travel 19.0 meters. We then measured the % of time-steps the robots were in formation after establishing the formation the first time (the distance travelled by the conductor up to this point is hereafter called *ft2*). As a supplement, we also tested the line, column, and circle formations (being the most 'extreme') with 6 robots. Results are shown in Tables 1 and 2.

$N=4$ 19.0 meters	*ft2*, meters	% time in formation after *ft2* (std.dev.)
line	0.7	88.4 (4.9)
column	3.0	100.0 (0.0)
diamond	2.6	100.0 (0.0)
wedge	1.8	92.0 (12.7)

Table 1: Stability, averaged over 10 runs for each formation.

$N=6$ 19.0 meters	*ft2*, meters	% time in formation after *ft2* (std.dev.)
line	3.1	84.3 (10.5)
column	5.8	100.0 (0.0)
circle	3.2	100.0 (0.0)

Table 2: Stability, averaged over 10 (9) runs for each.

The results for the line formation in Table 2 are averaged over only 9 runs; one run was a failure (for unknown reasons; the trials were not human-supervised). As can be seen in both tables, %'s are very high when no robots are required to look 90 degrees sideways; thus, the line formation turned out to be the most difficult. It seems that when the camera is panned sideways, the update rules tend to overcompensate somewhat. One remedy of this problem is to introduce a 'dead zone' around the correct spot, inside which the robot will not try to compensate [1]. This would decrease precision of the formation a bit, but improve stability.

For robustness, formation switching, and obstacle avoidance, the experimental results have similar structure: the robots establish the formation, the switch or the obstacle occurs after 6.0 meters, and the robots then re-form into an adapted formation. We tested these abilities by showing that 1) the initial formation is established (*ft2 (1)*) before the conductor has travelled 6.0 m, and 2) an adapted formation is established (*ft2 (2)*) before the conductor has travelled 19.0 m. Having demonstrated stability, we posit that for robustness, formation switching, and obstacle avoidance, it suffices to show that the robots do reach the adapted formation.

robot drop-outs (start: $N=6$)	*ft2 (1)* (std.dev.)	*ft2 (2)* (std.dev.)
circle, 1 out	3.2 (0.077)	8.9 (0.130)
diamond, 2 out	4.0 (0.205)	8.3 (0.235)

Table 3: Robustness, averaged over 10 runs for each.

We only report results from two **robustness** experiments (table 3), since the broadcasting of heartbeat messages (and hence the robustness of the algorithm) is very reliable. Once the conductor reached the 6.0 meter mark, 1 or 2 robots were eliminated. In all trials, the remaining robots successfully re-formed within the 19.0 m. For the circle, the conductor was euthanized. The remaining 5 robots realized this (no longer receiving the heartbeat from the conductor), promoted a new conductor, and eventually reached an adapted circle (a pentagon) after 8.9 m (see Figure 5). In the diamond case, the 6 robots reached the diamond the first time at 4.0 m – an incomplete '6-diamond'. At 6.0 m, the two perhipheral robots were terminated, leaving 4 the conductor (still the middle robot). Now, with only 4 robots remaining, a perfect 4-diamond was established by 8.3 m through a restructuring of the group.

Figure 5: A robustness run of 6 robots in a circle (a hexagon); one is terminated, and the rest adapt to a 5-circle, a pentagon.

Table 4 shows **switching** results. The line → diamond switch is averaged over 9 runs only; in one of the 10 runs, the robots failed to form the diamond in time, and thus for this run the *ft2 (2)* is void. All other trials were successful. When switching from the diamond to the wedge, the diamond has to open up in the back and 'unfold' (see Figure 6). The reverse happens when switching from the line to the diamond: the line folds back and the two extreme robots end up next to each other (their cameras panned in different directions). This is an example of a situation where letting the middle robot be the conductor minimizes congestion. If the leftmost robot in the line is an example of a switch from a centered to a non-centered formation, i.e., the conductor in the middle of the line steps down to let the left-most robot become the conductor of the column.

$N = 4$	*ft2 (1)* (std.dev.)	*ft2 (2)* (std.dev.)
diamond→wedge	3.6 (0.144)	8.3 (0.089)
line→diamond	1.1 (1.373)	9.7 (0.193)
line→column	4.0 (1.747)	9.2 (0.372)

Table 4: Switching, averaged over 10 (9) runs for each.

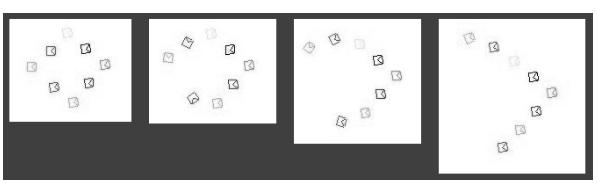

Figure 6: Example of eight robots performing a switch.

For **obstacle avoidance**, the four basic formations were tested with 10 trials each in a scenario where a wall (called $wall_1$), oriented orthogonally to the formation heading, was in the path of the formation. In addition, two more scenarios were tested with 10 trials each: a diamond formation crossing an obstacle field with 7 robot-sized obstacles, and a line formation being split by long, flat wall (called $wall_2$), parallel to the formation heading. Results are in Table 5.

Figure 7 shows a trial of a column evading $wall_1$. In all 10 trials of the diamond in the $wall_1$ scenario, the % of time in formation after $ft2$ was over 60 (not shown). In other words, after first establishing the diamond before

$N = 4$	*ft2 (1)* (std.dev.)	*ft2 (2)* (std.dev.)
line, $wall_1$	1.2 (0.900)	10.5 (1.081)
column, $wall_1$	3.8 (0.018)	13.1 (1.464)
diamond, $wall_1$	2.8 (0.070)	10.7 (1.274)
wedge, $wall_1$	2.1 (0.737)	11.8 (1.614)
diamond,obstacles	2.7 (0.143)	11.5 (0.925)
line, $wall_2$	0.7 (0.936)	14.7 (0.361)

Table 5: Simulation, obstacle avoidance; average over 10 (9) runs.

the obstacle, the robots managed to stay in formation while negotiating the wall in more than 60% of the time, due to a large *aheadbuffer*. In four trials, this number was 90% or more. For both sets of line trials, results are for 9 runs only; in one trial in each scenario, the robots did not re-form the line in time. All other runs were successful.

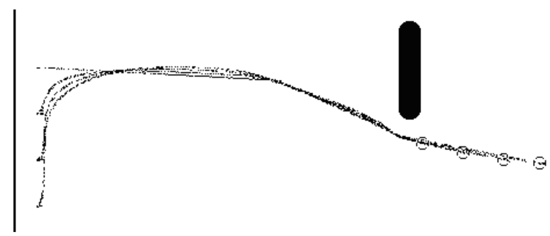

Figure 7: Simulation run: the conductor sights the obstacle ($wall_1$) from a distance, and makes an early swerve. Once past the wall, the conductor gradually returns to the original heading.

4.2 Physical Robot Experiments

The robot platform used for these experiments is the Pioneer2 DX from ActivMedia Inc. with sonars, the SICK LMS200 laser and the Sony PTZ Camera. We conducted formal real-robot experiments to assess stability and obstacle avoidance of our algorithm in the four basic formations, and informal tests with switching, but due to practical issues we only performed 5 trials of each experiment. For a full report of the experiment body, see [5].[3]

The experiments were carried out using four robots in a 3x5 meter arena. Due to the constrained space, we had the robots move very slowly (the default *tspeed* was 20 mm/sec) so that trailing robots could easily catch up, and they would have the time to get in formation within the arena. Results with real robots generally confirmed our findings from the simulated experiments. All formations with the exception of those that involve ±90 degree angles between friends, are very stable. Switching within centered formations went smoothly, and with 3 robots a switch from column to line was also successful. As for obstacle avoidance, our lab arena was not long enough for formations with more than two robots to take advantage of a large *aheadbuffer*, but with two robots keeping an incomplete diamond, obstacle avoidance in two scenarios with different types of walls was tested successfully (see figure 8).

[3] See http://robotics.usc.edu/~agents/projects/formations.html for video footage and more images.

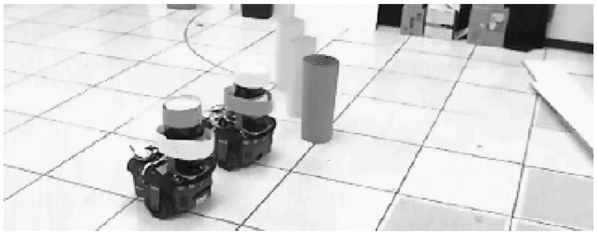

Figure 8: Sample image from obstacle avoidance.

Figure 9 shows overhead images from five robot runs: images a–d show stability runs of the basic formations, with robots moving from right to left (the starting line-up was to the far right in the correct order in almost a line formation; therefore there was not enough room to form a perfectly straight column). The sequence of images e–h shows a switch from diamond to line: In (f), the back robot is starting to pan its camera from -45 (looking left-ahead) toward +90 degrees (right) so as to get to the other side of its friend. The top robot is panning its camera from -45 degrees to -90 (left), and in (g), it has caught up with its friend. In (h), the line is completed with the trailing robot finding its place.

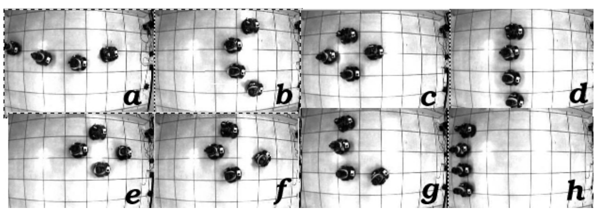

Figure 9: Example experiments on physical robots.

5 Conclusions And Future Work

The robots use only local sensing, but through simple communication they do have access to the global goal of the group: to do formation f with N robots. Hence, our system falls under the Type 1 category, according to [9]. We have validated our approach through extensive testing, both in simulation and on real robots. Our key concept for all formations is to follow a designated 'friend' robot at the appropriate angle and distance with respect to the desired formation, by using a panning camera, and thus simply keeping the friend centered in the image. This also makes it possible to switch easily between formations. The same algorithm could be used with any appropriate sensors for detecting the friend and the bearing to it.

Our algorithm proved very stable with friendship angles other than ± 90 degrees; for those, robots would occasionally move in and out of formation. A unique ID for every robot and a protocol for minimalist radio communication provide high robustness to drop-outs and late-comers, and help negotiate obstacles. A conductor that leads the way solves the problem of determining the friend's heading; by the nature of the algorithm, the only stable configuration is one in which robots eventually have the same heading as the conductor.

Recently we have been working on an improved version of the algorithm that offers much higher stability of the troublesome ± 90 degree friendship angles. Also, to further improve performance in the presence of obstacles, we are currently adding a few new message types to the communication protocol. E.g., when a robot encounters an obstacle ahead, it broadcasts this information along with its ID to help its neighbors – by the ID, they know who they are. With this augmentation, our system becomes one of Type 2 [9], where information about the route of the formation is available to all.

Notably, our work shows that no global positioning system is needed. With a global map and global knowledge of all robots' positions, more reliable performance is possible, but this paper demonstrates that even with only local sensing and control, a group of robots can display global, coordinated behavior in the form of stable, robust, switchable formations.

Acknowledgments

The authors are grateful to Richard Vaughan, Andrew Howard, Brian Gerkey, Boyoon Jung, and Esben Østergaard for valuable help with the simulator, LaTeX, geometry, etc.

References

[1] Tucker Balch, Ronald C. Arkin, *Behavior-based Formation Control for Multi-robot Teams*, IEEE Transactions on Robotics and Automation, Vol. 14, No. 6: 1–15, Dec. 1998.

[2] Tucker Balch, Maria Hybinette, *Social Potentials for Scalable Multirobot Formations*, IEEE International Conf. on Robotics and Automation (ICRA-2000): 73–80.

[3] Qin Chen, J. Y. S. Luh, *Coordination and Control of a Group of Small Mobile Robots*, IEEE International Conf. on Robotics and Automation (ICRA-1994): 2315–2320.

[4] Jaydev P. Desai, Vijay Kumar, James P. Ostrowski, *Control of Changes in Formation for a Team of Mobile Robots*, IEEE International Conf. on Robotics and Automation (ICRA-1999): 1556–1561.

[5] Jakob Fredslund, Maja J Matarić *A General, Local Algorithm for Robot Formations*, Institute for Robotics and Intelligent Systems Technical Report IRIS-01-396, University of Southern California, 2001.

[6] Brian Gerkey, Kasper Støy, Richard T. Vaughan, *Player Robot Server*, Institute for Robotics and Intelligent Systems Technical Report IRIS-00-391, University of Southern California, 2000.

[7] C. Ronald Kube, Hong Zhang, *Collective Robotic Intelligence*, Second International Conf. on Simulation of Adaptive Behavior (SAB-1992): 460–468.

[8] Maja J Matarić, *Designing and Understanding Adaptive Group Behavior*, Adaptive Behavior, vol. 4, 1: 50–81, MIT Press December 1995.

[9] Lynne E. Parker, *Designing Control Laws for Cooperative Agent Teams*, IEEE International Conf. on Robotics and Automation (ICRA-1993): 582–587.

[10] Richard T. Vaughan. *Stage: a multiple robot simulator*, Institute for Robotics and Intelligent Systems Technical Report IRIS-00-393, Univ. of Southern California, 2000.

[11] Barry Brian Werger, *Cooperation Without Deliberation: A Minimal Behavior-based Approach to Multi-robot Teams*, Artificial Intelligence 110(1999): 293–320.

[12] Barry Brian Werger, Maja J Matarić, *Robotic 'Food' Chains: Externalization of State and Program for Minimal-Agent Foraging*, From Animals to Animats 4: Proc. of the Fourth International Conf. on Simulation of Adaptive Behavior, eds. Maes, Matarić, Meyer, Pollack, and Wilson, 625–634, MIT Press 1996.