# **Robots in Formation Using Local Information**

Jakob Fredslund Computer Science Dept. Aarhus University Aabogade 34 8200 Aarhus N, Denmark chili@daimi.au.dk Maja J Matarić Interaction Lab, Computer Science Dept. University of Southern California University Park Los Angeles, CA 90089, USA mataric@cs.usc.edu

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. The goal is to have N mobile robots establish and maintain some predetermined geometric shape. We report results from 40 experiments with physical robots, showing the viability of our approach. The key idea is to keep a single *friend* at a desired angle by panning the camera and centering the friend in the image. We present a general analytical measure for evaluating formations and apply it to the position data obtained from two data gathering lasers tracking the robots during the experiments.

## 1 Introduction

This paper describes our continuing work [6] on the problem of achieving global behavior in a group of robots using only local sensing, with *formations* as an instance of that general problem. By *local* we mean that the robots do not know the position of other robots, except what they can sense themselves locally; in fact, in our algorithm each robot uses only the relative position of one other robot. 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 *global* quantitative criteria for evaluating the formations. In [6] we validated the algorithm largely through simulation; here we present an improved algorithm and results from applying this algorithm *and* the quantitative evaluation criteria to a group of four physical mobile robots.

## 2 Related Work

A variety of approaches have been proposed to create global behavior in a group of mobile robots. In [10], 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. With formations, however, a more rigid and reliable structure is needed from the group of robots. Each robot has to somehow determine its spot relative to the position of its peers. In [2], three ways of doing this are identified: *neighbor-referenced*, where the robot decides its position relative to one predetermined neighbor,

*unit-center-referenced*, where the robot references itself to the centroid of all robots, and *leader-referenced*, where the robot uses the position of a predetermined leader. 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, but this 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 the Leader-Following Control mode, one of the modes in a general high-level framework for programming multi-robot systems [1], each robot references itself to one neighboring robot, using only locally available information, maintaining a certain angle,  $\psi$ , and distance, l, to it. Thus, the needed information is position and orientation of one robot close by and within line of sight. An experiment with physical robots (though only two) is reported, where the follower robot keeps a pre-set heading and distance to the lead robot. The follower uses a camera and color-blob detection to identify the lead robot and its heading and distance. In [3], 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; however, 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., [4, 5].

There is thus a spectrum of strategies, ranging from simple, 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 1) global goals, and/or 2) of the actions and intentions of other robots. Within this framework, a robot knowing what formation and with how many robots it is supposed to participate in would be 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. She illustrates how the addition of global knowledge can improve system performance through formation simulations with four robots.

Of the two works that resemble our own the most, [2] does not demonstrate neighborreferenced formations with real robots using only local information, and [1] showed leader-following with two robots at a fixed angle. We demonstrate our formations with four real robots that dynamically change the angles they are keeping to their neighbor to switch between formations and adapt the formation if the group size changes.

### 3 Algorithm

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 our algorithm works for almost any geometric shape. Our key idea is this: every robot positions itself relative to one designated neighbor robot, its *friend*, using some appropriate *friend*-

sensor. To keep the algorithm for maintaining this position simple and general, the robot *pans* its friend-sensor some number of degrees pertaining to the current formation; thus, maintaining a place in the formation is simply keeping the friend in the center of the sensor's field of view – for all formations. Each robot has a unique ID number that it broadcasts reguarly as a heart-beat message; other robots can detect this ID. From the heart-beats, each robot knows how many robots are participating in the formation (N), and their IDs. One robot is the *conductor*, deciding the heading, and thus not following any friend (the term *conductor* is analogous to *leader* in the literature). All other robots follow a *friend*, and so all robots serve as "local leaders", and all are also followers (except the conductor). The conductor broadcasts a byte designating which formation to do, f, along with its own ID. This is an example of Type 1 global knowledge, as defined in [9]. The conductor does not broadcast its heading. Thus, the robots are organized in a *chain of friendships*, which is always kept sorted by ID. Since the conductor defines the formation heading, it should have a clear line of sight, and since it is the fix-point of the formation, it should be as much in the center as possible, so as to minimize congestion during formation switching. Therefore, for the *centered* formations (all except column), the robot with the middle ID (of those currently alive) is the conductor. For the column, the robot with the lowest ID is conductor, leading the formation. Hence, depending on N and f, any robot might serve the conductor duty.

This approach offers several nice implications. First, once the conductor starts moving, the only way for other robots to keep a stable position relative to their friend is by finding the friend's heading. In this way, the conductor 'drags' the whole formation into place just by going its own way. No global heading needs to be agreed upon, it is solved by self-organization. Since any robot can be the conductor, what seems a centralized element really is not. If the conductor fails, or if N otherwise changes, another robot can take 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 friend-sensor into the appropriate angle (*camangle*); the change in position results automatically.<sup>1</sup> Third, there is no global coordinate system and hence no communication of coordinates. The behavior-based controller consists of three concurrent behaviors and a module holding state data (Figure 1). Each is described in turn.

The state module WhatDoIKnow: Here resides all state information: the robot'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: This behavior receives the heart-beat messages from the other robots and maintains N and the table of IDs in WhatDolKnow. This information is used to calculate lessThanMe, the number of live robots with IDs lower than this robot's own ID (needed since IDs need not be consecutive). If N changes, the robot might be promoted to be the conductor. This happens if its lessThanMe value becomes equal to  $\lfloor N/2 \rfloor$ , in which case it is the middle robot.<sup>2</sup> Conversely, it could also be demoted if its lessThanMe is no longer  $\lfloor N/2 \rfloor$ . In this case it looks up a friend in the table of live IDs.

The channelCListener behavior: This behavior receives formation messages from the conductor and updates the formation variable, f, if necessary. If f changes, the robot might have to pan its camera to a new angle. The correct angle, camangle, is calculated from a simple geometric relationship of less ThanMe and N. For any N, the robots' respective camangles will result in a formation that is either uniform or incomplete; i.e., attempting a diamond with 5 robots will result in an incomplete 3x3 diamond, not

<sup>&</sup>lt;sup>1</sup>Switching to/from the column are special cases: they involve switching the conductor between the leftmost and middle robots, so the robots between the two find a new friend on the opposite side.

 $<sup>^{2}</sup>$ In case of the column, the front robot should be the conductor, and so the robot whose *lessThanMe* is 0 will get the promotion.

an overcrowded 2x2.

Also, *channelCListener* handles messages from other robots that have encountered an obstacle and are evading it. If a robot detects an obstacle in its path (see below), it will swerve to evade that obstacle and send out a warning, a *swerve* message, with its ID and a value indicating the turn angle and direction of the turn. Other robots, not necessarily sensing the obstacle, will react to this message by making a swerve of solidarity of equal turn angle and direction, if the sender is swerving their way (if they themselves sense an obstacle, their own avoidance behavior takes precedence. As the leader is in front, it will have a clear line of sight and thus should always resolve any deadlocks by just going its way, even-



Figure 1: The controller.

tually dragging other robots with it). This principle is documented in the  $O_1$  experiment reported in Table 1 below. Thus, robots share with each other knowledge about upcoming obstacles, categorizing our system as Type 2, following [9].

The main behavior: A robot R moves 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 heart-beat message, sets *tspeed* and *rspeed* to their default values (20 mm/sec and 0 deg/sec, respectively), 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 or made a swerve of solidarity. In this case it will gradually modify rspeed so as to return to the heading it had before the interruption. This heading is stored in WhatDoIKnow. If R is not the conductor, it will first locate its friend; R identifies its friend's ID by N, f, and the table of live IDs. If other robots are swerving its way to evade obstacles, R makes a swerve of solidarity. Otherwise, R will first get close to its friend: it does so by panning its camera straight ahead (yielding a straight path towards the friend) and speeding up. Once appropriately close, R will start panning the camera towards camangle, the right angle with respect to the current formation, and make small corrections to rspeed and tspeed so as to center its friend in the image. These suggested values of rspeed and tspeed are then given 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. Then, it is checked if any obstacles are found within this bounding box. If so, a correction is made to *tspeed* and *rspeed* proportional to the proximity of the sensed obstacle. Thus, *aheadbuffer* induces immediate collision avoidance. In addition, if it is set to a high value, it allows R to look far ahead for obstacles, resulting in an elegant, smooth avoidance behavior. 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, if R has been out of place for a long time, or if R has other robots in front of it when moving in the formation (derived from N, less ThanMe and f). Otherwise, aheadbuffer is set to a higher value proportional to robot-size and N; a large formation needs more space to negotiate obstacles than a small one. A higher aheadbuffer gives more time to react, and more time yields more space. 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.

#### 4 Experimental Evaluation

Having reported extensive simulation results in [6], here we focus on data obtained from trials with four physical robots. We used ActivMedia Inc. Pioneer2 DX robots with the SICK



Figure 2: Switching from diamond to line.

LMS200 laser, sonars, and the Sony PTZ camera, running *Player* [7], a server and protocol that connects robots, sensors, and control programs across the network. The camera (with color-blob detection software) functioned as the friend-sensor; each robot wore a customized helmet with two fluorescent color stripes, identifying its ID to other robots. Figure 2 shows overhead images of a switch from diamond to line.

Since the robots share no global coordinate system, and the field of view of the overhead camera was too small, we had to find an observer's way of obtaining position data from our experiments. We used a tracking system, written by Andrew Howard (see robotics.usc.edu/player): two lasers monitored the four robots, detecting special reflective beacons mounted on top of the helmets. To be tracked, a robot had to be visible to at least one of the data gathering lasers, but occlusions occasionally occured, especially when switching between formations. As our lab has six lasers in total, only two were available for data gathering.

We propose the following *formation evaluation criteria* as a means of judging quantitatively the notion of being in formation:

<u>Definition1</u> 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 'stretch 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 perhaps adjust the angles a little, so that all robots are close to this tweaked shape: no angle in the original shape must be stretched more than  $\epsilon_a$  to make the data points fit. Criterion 3 states that the stretching from criterion 2 must not skew the heading too much. Note that by using 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 in the sense that N robots are *not* considered to be in line even if the angular offset between neighboring robots is small, if in fact overall they form, say, an arc. I.e., it is not enough that all robots *locally* are keeping more or less the right angle to their friends; criterion 2 ensures a *global* quality of the group.

Our assumption is that robots start out in the right order with respect to the chain of friendships but not necessarily with their respective friends in the visual field. This is reasonable, given that the problem of aggregating robots into such a formation has already been empirically demonstrated [8]. Furthermore, following ideas from [11], 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 helmets with IDs could be communicated. Our experiments were designed to document stability, robustness, obstacle avoidance, and switching between formations.

To show stability, the robots were first placed close to the desired formation. Using Definition 1, we then recorded the % of time they were in formation after establishing it for the first time (ft2). To justify this, we also did experiments showing that the robots could actually get into any formation from any initial configuration, as long as they started out in the right order.

Table 1: Experiments with 4 robots, five trials each. Numbers are % of time in formation after ft2, using Definition 1. O<sub>1</sub> and O<sub>2</sub> are obstacle avoidance experiments with 2 robots.

Diamond	Wedge	Line	Column	<b>O</b> <sub>1</sub>	$O_2$
74.96	100.00	94.05	96.11	100.00	81.77
99.68	99.87	46.80	100.00	97.89	81.97
88.60	99.56	70.43	98.87	99.82	89.39
99.80	50.32*	34.71	100.00	97.80	21.00
100.00	49.76	64.88	100.00	93.92	$26.73^{\dagger}$

Table 1 displays a summary of our large body of collected tracking data (in total, we performed more than 40 real robot experiments). Definition 1 was used as evaluation measure with desired inter-robot distance  $d_{dist} = 80$  cm,  $\epsilon_{d_1} = (0.20 * d_{dist})$ ,  $\epsilon_{d_2} = (0.08 * d_{dist})$ , and  $\epsilon_a = (0.08 * 2\pi)$ . In other words, the dispersion of robots was set fairly loosely, allowing actual inter-robot distances to differ with up to 20% of  $d_{dist}$  (see Section 5 for a discussion of real-world influence on precision). Criterion 2 was applied more strictly: all robots had to be at most 8% of  $d_{dist}$  from the line segments they belonged to<sup>3</sup>. Angles were allowed to deviate 28.8 degrees. We used a simple line-fitting algorithm to fit data points to straight lines.

The stability of the four basic formations is shown in the four leftmost columns of Table 1. Column and Diamond are indeed very stable. Wedge seems to be either very stable or rather unstable, and the line can hardly be dubbed anything but unstable. These facts stem from two related problems.

Wedge and line share the feature that some robots have to pan their cameras  $\pm$  90 degrees to look for their friend. This orientation proved to be the most difficult to maintain; it is very hard for a robot R to realize if it is ahead of or behind its friend F, if their headings are slightly different. If R thinks it is ahead, it will speed up, possibly

<sup>&</sup>lt;sup>3</sup>For the line, we set  $\epsilon_{d_2} = (0.30 * d_{dist})$ . Otherwise, at least one robot would often find itself too far away from the best-fit line.

resulting in an unstable, oscillatory course. An example of this (the run marked with a \* in Table 1) is shown in Figure 3(b). The other problem is that the laser range is only  $\pm 90$  degrees: if R is right next to F (when its camera is panned  $\pm 90$  degrees), F will be at the boundary of R's laser range. If F drops beyond that boundary, R can no longer judge the distance to F accurately, and so again an oscillatory course can result.

In the  $O_1$  experiment of Table 1, two robots formed an incomplete diamond, i.e., the follower, R, kept a 45-degree angle to the conductor, C. They had to negotiate a wall that was only in R's path. By the combined workings of the long *aheadbuffer* and the *swerve messages* sent out by R, C made swerves of solidarity in time to let Rkeep its position. Hence the robots maintained the formation while still evading the wall, as seen by the high %'es of experiment  $O_1$  in Table 1. Due to lack of space, it was not possible to show global formation obstacle avoidance with more than 2 robots.

In  $O_2$ , the wall was in both of their paths, but this time the wall had a passage in the middle, not wide enough to allow the robots to pass it while in formation. As seen by the three still high %'es in Table 1, only for a short time did R lose its position before regaining it. The run marked with  $\dagger$  is shown in figure 3(c); here R did not follow C through the passage in the wall, but instead went around it to the left, completely losing sight of C. Generally in this situation, R will attempt to get back to the heading it had when it was last in position, and so here it made a right curve tightly following the wall. Eventually it spotted C in the distance and managed to catch up and get into position for last 100 time steps. The 21%-run also demonstrated a recovery where Rre-established the formation before the last 100 time steps.

In Figure 3(a), a switch from diamond to wedge is shown: when given the command to switch, the back robot of the diamond pans its camera to get its friend on its right rather than its left, sliding behind the others and ending up in place (compare with Figure 2). The robots tended to occlude each other during switching, so it was very hard to get



Figure 3: (a) Switching from diamond to wedge. (b) An unstable wedge. (c) Two robots split around an obstacle and then re-join.

tracking data, but by human inspection (not by the authors), the robots were reliably capable of switching between any centered formations, and between line and column. By the heart-beat messages, each robot knows who else is alive. That introduces robustness into our system, so that if a robot fails, the formation will adapt, and if necessary, a new robot will become the conductor. Since this works extremely reliably, both due to reliable (minimal) radio communication and reliable color helmet detection, we do not report robustness experiments here (see [6] for simulation data).

#### 5 Conclusions

Our robots use only local sensing (i.e., they each know only the (relative) position of one other robot), and through simple communication they know the global goal: to do formation f with N robots. Also, robots inform each other of upcoming obstacles. Hence, our system is of Type 2, according to [9]. Our key concept is to follow a designated 'friend' robot at the appropriate angle and distance (similarly to [1, 2]), by using a panning camera, and thus simply keeping the friend centered in the image. This also enables easy switching between formations. Unique IDs and a protocol for minimalist radio communication provide robustness to drop-outs 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 when all robots eventually have the same heading as the conductor.

Our method proved highly successful for certain formations (diamond, column, and, to some extent, wedge), but our expectations of problems with the line were also confirmed. We have since, however, improved the algorithm and almost eliminated the oscillatory behavior described above. We validated the algorithm through 40+ experiments with physical robots<sup>4</sup>. Many real-world issues affect the efficiency of the algorithm. E.g., since the physical shape of the robots is not completely symmetric, when a robot perceived a distance of n millimeters to its friend, its center was usually not at distance n from its friend's center. Further, the extent of this error varied with the angle between the robots. Finally, however well our colored helmets worked, they were not perfect: depending on ambient light and perception angle, the center of their color stripes was not always aligned with the camera, thus introducing a small error in the angular positioning of a follower robot to its friend.

We believe our results show that having a working multi-robot system in simulation does not necessarily prove that the algorithm will work with real robots; many and varied experiments with several robots should be performed.

#### References

- R. Alur, A. Das, J. Esposito, R. Fierro, G. Grudic, Y. Hur, V. Kumar, I. Lee, J. Ostrowski, G. Pappas, B. Southall, J. Spletzer, C. J. Taylor, A Framework and Architecture for Multirobot Coordination, International Symposium on Experimental Robotics (ISER 2000), Hawaii, December 10-13, 2000
- [2] Tucker Balch, Ronald C. Arkin, Behavior-based Formation Control for Multi-robot Teams, IEEE Transactions on Robotics and Automation, Vol. 14, No. 6, pp. 926–939, Dec. 1998.
- [3] Tucker Balch, Maria Hybinette, Social Potentials for Scalable Multirobot Formations, IEEE Intl. Conf. on Robotics and Automation, Vol. 1, pp. 73–80, San Francisco, 2000.
- [4] Qin Chen, J. Y. S. Luh, Coordination and Control of a Group of Small Mobile Robots, IEEE Intl. Conf. on Robotics and Automation, pp. 2315–2320, San Diego, 1994.
- [5] Jaydev P. Desai, Vijay Kumar, James P. Ostrowski, Control of Changes in Formation for a Team of Mobile Robots, IEEE Intl. Conf. on Robotics and Automation, pp. 1556–1561, Detroit, May 1999.
- [6] Jakob Fredslund, Maja J Matarić, Robot Formations Using Only Local Sensing and Control, IEEE Intl. Symp. on Computational Intelligence in Robotics and Automation, pp. 308–313. Banff, Alberta, Canada, July 29 - August 1, 2001.
- [7] Brian P. Gerkey, Richard T. Vaughan, Kasper Støy, Andrew Howard, Gaurav S. Sukhatme, Maja J Matarić, Most Valuable Player: A Robot Device Server for Distributed Control, To appear: Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, Wailea, Hawaii, October 2001.
- [8] Maja J Matarić, Designing and Understanding Adaptive Group Behavior, Adaptive Behavior, vol. 4, 1:50–81, MIT December 1995.
- [9] Lynne E. Parker, Designing Control Laws for Cooperative Agent Teams, IEEE Intl. Conf. on Robotics and Automation, pp. 582–587, Atlanta, May, 1993.
- [10] Barry Brian Werger, Cooperation Without Deliberation: A Minimal Behavior-based Approach to Multi-robot Teams, Artificial Intelligence 110 (1999), pp. 293–320.
- [11] Barry Brian Werger, Maja J Matarić, Robotic 'Food' Chains: Externalization of State and Program for Minimal-Agent Foraging, From Animals to Animats 4: Proc. Fourth Intl. Conf. on Simulation of Adaptive Behavior, eds. Maes, Matarić, Meyer, Pollack, and Wilson, pp. 625–634, Cape Cod, MIT September 1996.

<sup>&</sup>lt;sup>4</sup>See http://robotics.usc.edu/~agents/projects/formations.html for video footage.