Energy Efficient Robot Rendezvous

Pawel Zebrowski, Yaroslav Litus, Richard T. Vaughan
Autonomy Lab, School of Computing Science
Simon Fraser University, Canada
(pzebrows, ylitus, vaughan)@sfu.ca

Abstract

We examine the problem of finding a single meeting location for a group of heterogeneous autonomous mobile robots, such that the total system cost of traveling to the rendezvous is minimized. We propose two algorithms that solve this problem. The first method computes an approximate globally optimal meeting point using numerical simplex minimization. The second method is a computationally cheap heuristic that computes a local heading for each robot: by iterating this method, all robots arrive at the globally optimal location. We compare the performance of both methods to a naïve algorithm (center of mass). Finally, we show how to extend the methods with inter-robot communication to adapt to new environmental information.

1 Introduction

If a team of robots operates in some environment it is easy to imagine a situation when all robots should assemble in one place: a *rendezvous*. This rendezvous could be an initial step in formation creation, or it could be required for robot recharging, maintenance, or collection. We are particularly interested in long term autonomous robot operations, in which energy availability is a critical factor. Therefore we are concerned with the energy efficiency of all robot tasks, including essential self-maintenance. Previously we have proposed a scheme whereby a mobile *tanker* robot recharges *worker* robots by docking with them: a recharging rendezvous [18]. In this paper we describe practical algorithms which solve the problem of finding a single rendezvous point for multiple robots, each with independent locomotion costs, such that the total system energy cost of traveling to the rendezvous is minimized.

The problem of path planning for multiple robots to assemble in one location, the *rendezvous problem*, has already received some attention. Most authors consider the setting where robots have incomplete information about locations of each other which makes it difficult to coordinate and agree on a single meeting point [4,5,8,15]. For example, [1] describes and proves a distributed rendezvous algorithm for robots with limited visibility. Other authors concentrate on selecting a meeting point to ensure that some specific properties hold, or to optimize the formation during the convergence. [10] presents an algorithm for finding the meeting point which minimizes the maximum individual travel costs to a single meeting point on a weighted terrain. [16] describes a scheme which makes the convergence process more organized in a certain mathematical sense.

In this paper we assume perfect shared knowledge about the robot locations and show an effective algorithm which minimizes total energy spent by the robot team in performing their rendezvous. This problem is different from that solved by [10], who instead minimized the maximum energy cost to any individual robot. Our scheme optimizes *energy efficiency*, which is critical for some applications.

2 Problem Characterization

Assume n robots are located at positions r_i, $i = 1 \ldots n$. When a robot moves, it expends energy proportional to the length of its trajectory. Robots have individual energy costs c_i per unit of traveled distance, thus if robot i moves from a to b, it spends $c_i||a - b||$ units of energy. Now the task is to find a point p^* which minimizes the total energy spent by all robots for meeting at that point:

$$p^* = \arg\min_p \sum_{i=1}^{n} c_i||p - r_i|| \tag{1}$$

Problem (1) is known very well in optimization, where it belongs to the family of *facility location* problems. Historically, it has a variety of names including the Fermat-Steiner problem, Weber problem, single facility location problem, and the generalized Fermat-Torricelli problem. Though it is not possible to find a closed-form solution for this problem, the properties of its solutions are well known (see [7] for the case $n = 3$ and [9] for the general case). Effective numerical algorithms exist [14]. Interestingly, it is also possible to describe the solution using a mechanical interpretation as a system of idealized strings, pulleys and weights [13].

We will now briefly state the properties of the solution point. First, p^* obviously can not be located outside the convex hull formed by r_i. Second, if points r_i are not colinear (not lying upon a straight line), the goal function in (1) is strictly convex, which ensures the uniqueness of p^* if $n \geq 3$. Finally, it is possible to prove the following theorem [9].

Theorem 1 *If r_i are not colinear and for each point r_i*

$$||\sum_{j=1}^{n} c_j \frac{r_j - r_i}{||r_j - r_i||}|| > r_i, i \neq j \tag{2}$$

then $p^* \neq r_i$ for any i and $\sum_{i=1}^{n} c_i \frac{p^* - r_i}{||p^* - r_i||} = 0$ (the floating case). If (2) does not hold for some r_i then $p^* = r_i$ (the absorbed case).

Intuitively, in the floating case all robots meet at some point which is not the starting point of any robot: all robots move. In the absorbed case, one robot stays still and the others drive to it.

If r_i are colinear, then there may exist more than one point where minimum energy cost is incurred and this method may fail. This is not a significant practical problem for two reasons. First, in most real world situations, the robots are unlikely to be perfectly aligned (with the exception of 1-degree of freedom robots such as trains). Second, even in the colinear case the "local dynamic" method presented below converges to a unique instance of the possible minima.

3 Solution

3.1 Global Method

Since in our setting robots know each other's locations, the straightforward way to rendezvous is for each robot to find p^* using a numerical method and then move towards it. Alternatively, if communication is possible, one robot can calculate p^* and broadcast it to the other robots. We will call this approach the *global* method because it computes a single point in global world coordinates common to all robots. The global method provides a complete solution to the problem provided the complete traversal costs are computable in advance, i.e. a complete map of obstacles is available in advance, and robot travel costs do not change. Any changes to travel costs that are detected during run-time will require a complete recalculation to ensure the best solution. In the experiments below, we will refer to the global method using only the static initial conditions as the *global static* method. If the global method is recalculated to take into account new information, we call this the *global dynamic* method.

In the next section we propose a fast local heuristic method which allow robots to move towards p^* without ever calculating its location. An advantage of this method is that, in its dynamic form it naturally adapts to run-time changes in robot travel costs.

3.2 Local Method

In this heuristic *local* method, each robot's physical trajectory approximates a gradient descent towards point p^* on the total movement cost function adaptive landscape. Since in the general case the gradient of the cost function is *not* pointing directly at p^*, the gradient must be reevaluated at suitable intervals. Each robot moves in the direction of the local gradient, which is periodically recalculated, until all robots arrive approximately at the rendezvous point. At no point is the complete landscape or a complete trajectory computed.

This method can be considered an example of the "information surfing" technique described by Bourgault et.al. [3]. Our contribution is the design of novel algorithms that approximate the minimal global energy cost objective function, and the empirical evaluation that suggests this is a practical solution to the rendezvous problem.

First, we introduce the function which returns a unit length vector in the direction from point a to point b:

$$\vec{d}(a, b) = \frac{b - a}{||b - a||} \tag{3}$$

We present two versions of this algorithm. Each robot runs the same algorithm asynchronously in parallel. The *local static* method uses only the initial robot locations and fixed movement costs. The *local dynamic* method assumes that robots periodically broadcast their current position estimate as they drive; the algorithm uses the latest position estimate for each robot. The two methods have different costs and benefits, which we describe below.

Algorithm 1: Local Static

1. Update current location of self, x.
2. Calculate $\vec{D} = \sum_{i|x \neq r_i} c_i \vec{d}(x, r_i)$.
3. If $x = r_i$ for some i, set $c = c_i$, otherwise set $c = 0$.
4. If $||\vec{D}|| < c$ then stop. Otherwise proceed in the direction $\vec{D}$.
5. Goto step 1.

Algorithm 2: Local Dynamic

Let self be the robot originally located at r_j

1. Update current location and movement cost of self, r_j, c_j. If information about other robots was received, update it as well.
2. Let $A = \{i, \text{where}||r_i - r_j|| \leq \epsilon\}$, meaning the set of robots which are closer to r_j than some threshold ϵ. Note that $j \in A$, thus A has at least one element.
3. Calculate $\vec{D}_j = \sum_{i \notin A} c_i \vec{d}(x, r_i)$.
4. Set $c = \sum_{i \in A} c_i$.
5. If $||\vec{D}_j|| < c$ then stop. Otherwise proceed in the direction $\vec{D}_j$.
6. Broadcast own position and movement cost.
7. Goto step 1.

Both versions of the local method converge to the optimal meeting point (or any one of the optimal points in the co-linear case). Due to the lack of space we omit the complete proof of convergence. The idea of the proof is seen in the fact that vectors D and D_j point in the direction of the conjugate gradient of the total energy cost function. Thus, the current positions of the robots serve as current solution approximations in a first order numerical gradient descent optimization of the cost function. The absorbed case is accounted for by the stopping condition. Convergence of the first order method is proven in [17]. We do not expect much gain in calculating the movement vector using second order approximations and believe that more complex calculations will not yield worthwhile reductions in total trajectory length. However, exploring this option may be one further direction of this research.

The local approach has important benefits in comparison with global methods. First, if the convergence trajectories performed using the local method are only slightly longer than the optimal straight paths to p^* (as shown empirically below), then the robots may meet a little sooner since they do not need to wait until the calculation of p^* is over to start moving. Second, if conditions change during the progress of convergence, the local method will incorporate changes instantly, adapting to the new information without the need for a computationally expensive recalculation of the meeting point. Conditions may change for several reasons, for example:

1. A robot may need to quit the rendezvous routine or a new robots may enter.
2. A robot may pick up additional load which will make it more costly to move (or the inverse).
3. A robot may need to depart from its trajectory towards the meeting point because of environmental obstacles. In the extreme case the meeting point calculated using the original locations of the robots may end up *inside* a newly-discovered obstacle, and a replacement meeting place must be found.

Any of these events could cause the optimal meeting place to move significantly. This means that, for the global methods, a complete solution must be computed from scratch. In the last example, a robot may encounter new pieces of an obstacle it is navigating around, causing a stream of recalculations that may produce useless meeting points that turn out to be inside yet more obstacles. In contrast, the local methods quickly compute only the direction to move *right now*, that is towards the current best meeting place.

4 Experiments

A set of experiments is performed to demonstrate and compare the proposed global and local methods. For further comparison, a naïve method in which the robots meet at their center of mass is also tested. Each of these three methods is tested in its static and dynamic variants, for a total of six experiments.

4.1 World and Simplifying Assumptions

Experiments are performed in simulation using the well-known Player/Stage robot control and simulation system [6]. The world is an empty circular arena 40 meters in diameter, containing ten mobile robots modeled after the ActivMedia Pioneer 3-DX with SICK LMS-200 laser range finders. Each robot is assigned an individual weight c_i which describes its energy consumption per unit distance traveled. Total energy used as robot i moves from a to b is assumed to be $c_i||a - b||$.

During the experiment, robots are not visible to each other nor can they collide with each other. This unrealistic model is intentionally chosen in order to eliminate the effects of inter-robot spatial interference in this study. Spatial interference is a significant issue in multi-robot systems, particularly when the robots *must* operate in the same region, and rendezvous is the extreme case. Spatial interference is the critical limiting factor in how closely robots can approach the ideal rendezvous point, and will strongly determine the design of stopping conditions for any rendezvous algorithm. We recognize the importance of this topic [19], but believe it can usefully be ignored here to examine pure rendezvous performance, where our robots are treated as points that do not interfere with each other. Further, robots are assumed to have perfect localization, again to avoid influencing results based on the details of any one localization technique. The impact of localization error on the stability of our methods is an interesting area for future study.

4.2 Task

Given some initial arrangement of robots in the environment, the task is to meet at the unique location that minimizes the total system energy used on locomotion. Robots are deemed to have successfully met if for any two robots r_m and r_n the distance between them is less than some threshold d. In these experiments d is 1 meter.

The metric used to evaluate the performance of each method is the total system energy E used to achieve the rendezvous, where

$$E = \sum_{i=1}^{n} c_i d_i \tag{4}$$

and d_i is the length of the trajectory of robot i.

4.3 Controller Implementation

The experiment depends on two controllers: a robot controller and a processing controller. Communication between controllers is implemented with UDP datagrams. The next sections describe the controllers.

4.4 Low-Level Robot Controller

The robot controller is responsible for controlling robot movement within the environment. It reports the robot's current position to the processing controller, waits for the controller to prescribe the next move, then moves the robot to this location while avoiding obstacles. Obstacle avoidance uses the standard Player implementation of Borenstein's Vector Field Histogram method [2].

4.5 Processing Controller

A single centralized processing controller is used to offload computation from the individual low-level robot controllers. It tracks each robot's position and computes each robot's next move using the chosen method. In static methods only the robot's original starting positions are considered. In dynamic methods only the most recently received robot positions are considered.

4.5.1 Global Rendezvous Method

This controller implements the algorithm described in 3.1. Given the location of each robot r_i it is possible to construct a cost function for meeting at any location p in the environment. It is then possible to find the minimum of this function, as described by equation 1. This method achieves this using the Nelder-Mead algorithm [12]. Nelder-Mead returns the minimum cost location p^*. The processing controller then prescribes this as the location each robot should go to. In the dynamic variant p^* is recomputed periodically using the latest robot positions. In the static variant, the minimum cost location p^* is computed exactly once.

4.5.2 Local Rendezvous Method

The local method works by computing the conjugate gradient and prescribing the direction each robot should move in based on this gradient. In both static and dynamic variants, the local gradient at the robot's current position is recomputed periodically. The static variant implements Algorithm 1 above. The dynamic variant implements Algorithm 2 above.

4.5.3 Center-Of-Mass Method

This method calculates the center of mass given the arrangements of robots. The center of mass is simply the weighted vector sum of the robot positions. This point is then prescribed as the meeting place for each robot to move to. In the static variant, the rendezvous point is calculated exactly once from the initial robot positions. In the dynamic variant, the meeting point is recalculated periodically using the latest robot positions.

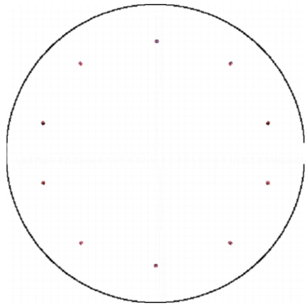
4.6 Experimental Consistency

Each experiment uses an identical low-level robot controller. Only the high-level method for selecting the next robot goal position is changed between trials. To avoid any artifacts introduced by the presence or absence of communication overhead, the static experiments actually perform communication but simply discard the received messages.

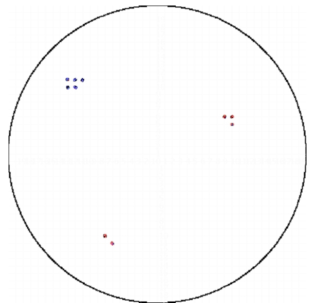
4.7 Procedure

Each of the six experiments is performed with seven distinct initial configurations illustrated in Figure 1. A configuration consists of a map that specifies each robot's starting position and orientation, and the position of any environmental obstacles. A configuration also specifies each robot's locomotion cost weight. Configurations 3 and 4 use the same start locations but different weights. Maps 6 and 7 incorporate an obstacle to demonstrate each method's ability to cope with obstacles.

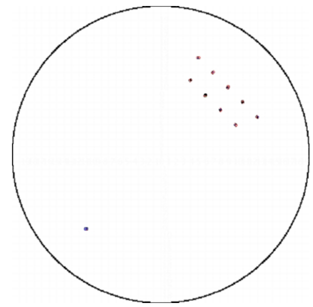
20 trials are performed on each configuration/method pair, for a total of $6 \times 7 \times 20 = 840$ trials. The total energy used in the trial is recorded and used to compute a mean



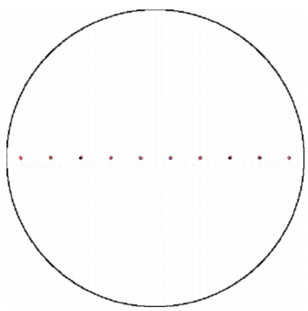
(a) Map 1 - All robots equal weight



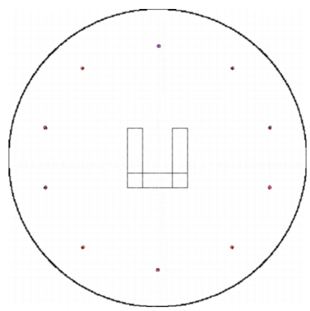
(b) Map 2 - All robots equal weight



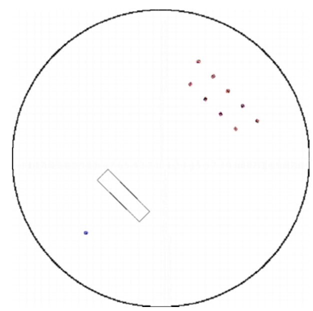
(c) Map 3 - Single robot weight = 10, all others weight = 1. Map 4 - Single robot weight = 1, all others weight = 10



(d) Map 5 - All robots equal weight

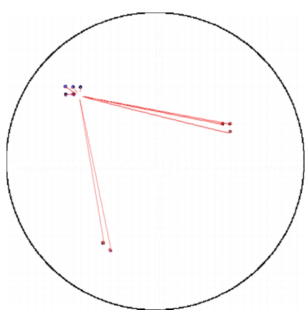


(e) Map 6 - All robots equal weight

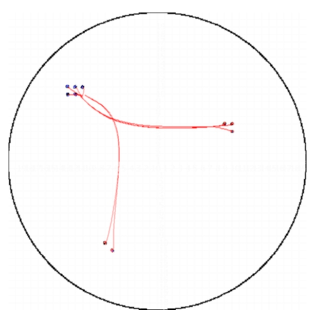


(f) Map 7 - All robots equal weight

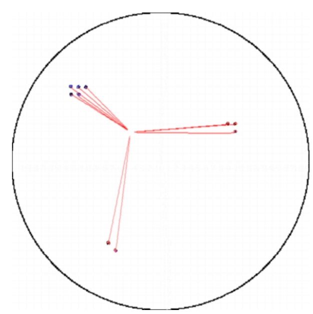
Figure 1. Experimental initial conditions



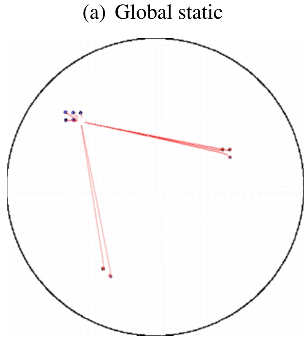
(a) Global static



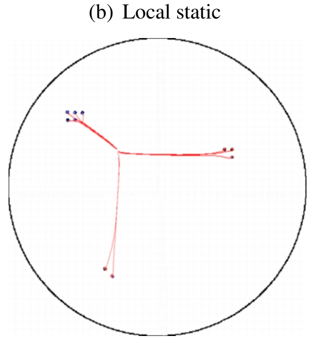
(b) Local static



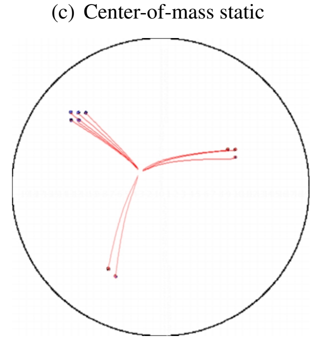
(c) Center-of-mass static



(d) Global dynamic



(e) Local dynamic



(f) Center-of-mass dynamic

Figure 2. Typical paths taken (Map 2)

and standard deviation. The path taken by each robot is also recorded. There is a time limit on experiments (ten times the typical trial length) in case a trial fails to result in a rendezvous.

5 Results

5.1 Paths Traversed

Figure 2 shows some example paths taken during the experiments. In each case, the global methods result in robots heading directly for the meeting location (Figure 2(a), 2(d)) while the local methods tend to take a curved path (Figure 2(b), 2(e)) as predicted in Section 3.2.

In the absence of obstacles (and spatial interference), all methods perform similarly regardless of whether the static or dynamic variant is used. The only exception is the local static method with colinear initial conditions, which fails to rendezvous as mentioned in Section 2.

In an environment with obstacles, all methods manage to achieve a meeting regardless of static or dynamic variant.

5.2 Energy Used

Table 1 shows the mean total energy used for each method/map combination. In each case without obstacles (Map 1-5) the global methods use less energy than the local methods. In cases with obstacles, this is not true. In cases where the center of mass coincides with the minimum energy cost location (as expected e.g. in Map 1), the center of mass method performs well, otherwise it performs poorly compared to the other methods.

6 Discussion

6.1 Paths Traversed

Figure 2 shows typical paths taken for each method using Map 2 (Figure 1(b)). Figures 2(a) and 2(d) show the robots meeting at approximately the optimal location. Figure 2(b) shows the robots also meeting at the optimal location, but the path traversed is an arc. This curve is produced by the robot following the local conjugate gradient. The local methods almost always result in curved paths that are slightly longer than those produced by the global methods.

In practice the cost of the longer path is traded off against the cost of the increased computation that may be required by the global method. It is unknown whether a robot that sits idle while computing the optimal meeting location then heading directly for it will consume less energy than a robot that begins moving immediately, but follows a longer path. This is a possible area of future research.

Figure 2(c) shows robots meeting at the center of mass. Notice that in this case, the center of mass does not coincide with the optimal meeting point and the energy usage is higher than the optimal methods. In the dynamic variant (Figure 2(f)) this is more pronounced, as the meeting point moves further away from the optimal location.

6.2 Static Versus Dynamic

The effect of dynamic updating can be seen clearly by comparing Figures 2(b) and 2(e). Updating position information results in robots “sticking” together when they meet (Figure 2(e)). This is a desired result, since two robots r and s at the same location are equivalent to one robot of weight $r + s$. Further, each robot in such a group should conclude the same shortest path to the meeting point. In the comparison of 2(b) and 2(e) dynamic updating achieves this by shortening the curved path caused by the local method.

Dynamically updating robot positions can act as a benefit or hinderance. In our trials, it results in both improved and degraded cases. The degradation tends to be minimal while improvements tend to be significant. One particular example of this is the colinear case using the local method, where without dynamic updates the robots never meet. In our trials, the naïve center of mass method never benefits from dynamic updates.

In an environment with obstacles, benefits of dynamically updating position information depend on the details of the environment. For example, in cases where an obstacle results in a shift of the optimal location, such an update is an asset. In cases where one obstacle results in a shift, then another results in a shift back, dynamic updating results in cyclic behaviour (and therefore higher energy usage). This is due to updates not yielding any real insight into the path that will need to be traversed, and is the expected result. However, one notable benefit of dynamic updating occurs when the meeting location chosen lies in an unreachable space (such as inside an obstacle). With the static methods, robots will perpetually attempt to reach an unreachable space. With the dynamic method, there is a chance that the optimal meeting location will shift away from the unreachable location. An algorithm which considers obstacles when computing the meeting location would be more suitable in this case.

6.3 Global Versus Local

Both methods achieve their intended goal: to identify a meeting location that can be reached at a minimum total cost. However, the methods exhibit different characteristics which make each more suitable in certain applications.

The global method performs well, but is computationally expensive. For a large number of robots computation may

Table 1. Mean Total Energy Used In Maps Without Obstacles. All Standard Deviations $< 5\%$

	Dynamic			Static		
Map	C-o-mass	Global	Local	C-o-mass	Global	Local
1	149.93	149.99	151.56	149.92	149.92	152.79
2	119.78	105.21	115.21	113.79	105.25	119.00
3	154.32	77.56	80.49	86.12	77.28	81.48
4	923.74	477.89	503.18	531.24	476.97	530.11
5	95.58	97.36	97.98	95.54	95.51	Failed

Table 2. Mean Total Energy Used In Maps With Obstacles. All Standard Deviations $< 6\%$

	Dynamic			Static		
Map	C-o-mass	Global	Local	C-o-mass	Global	Local
6	293.58	288.56	256.75	284.46	283.23	256.25
7	1150.64	533.58	549.57	579.20	5.24.22	600.02

become prohibitive. This is especially true in the dynamic case. Every position update results in a recomputation of the optimal meeting location. On the other hand, the local method is computationally cheap, but does not yield a location but rather a direction to head in. Further, the path traversed tends to be slightly longer than that of the global method. The suitability of each method depends on the application. For small teams of robots, the global method should perform well, even the dynamic variant. For larger teams, the local method will avoid lengthy computation before moving. If a meeting is desired in a minimum amount of time, it may be the case that computation time with the global method exceeds the extra travel time resulting from the local method. In such a case, the local method would result in a rendezvous before the global method, despite the longer path.

7 Conclusion and Future Work

We have stated the natural *robot rendezvous problem*, where multiple robots must meet at some location, chosen to optimize some utility function over all robots, and identified this as the single-facility location problem. We implemented a standard numerical solution and empirically evaluated its performance in a variety of scenarios. These data were then used to compare with the performance of a novel behavioural heuristic method, which was shown to (i) produce global rendezvous; and (ii) incur travel costs only slightly greater than the global optmization method. The novelty here is in designing algorithms 1 & 2 that create a local gradient that approximates the direction of the global optimum. These algorithms are very fast, with runtime growth linear with population size, and small constant per-robot cost. For some applications this may be preferred to an iterative numerical approximation technique with unknown runtime. Further, by iterating the local algorithm as the robot drives, we can naturally incorporate new information about robot locations, and thus cope with obstacles, robot locomotion failures, etc. without invalidating previous computation.

Some limitations of this work so far are in our assumptions of good global localization and reliable communication. The assumption of no robot collisions is not important, as this can be considered equivalent to the presence of obstacles, with which this method is shown to cope. We omit robot-collision experiments for brevity.

Consideration of the local heuristic method suggests that in the presence of unbiased error in global localization, these methods will cause the robots to converge until their mutual distances are within a small factor of the mean localization error. Given that global localization methods that give accuracy to less than a robot's sensor range are in everyday use, our method should be practical.

As for unreliable communications, we believe that occasional dropped messages will cause only small changes in individual robot behaviour, leading to graceful degradation in overall performance. On the other hand, a robot that loses communications completely during a run may be unrecoverably lost if the other robots encounter obstacles that cause the optimal rendezvous point to change.

Some opportunities for future work are identified in the relevant sections above. We have recently published a solution to the related (and harder) problem of energy-efficient multi-robot rendezvous where multiple meeting locations are allowed [11]. We aim to use the methods described in these papers in our real multi-robot system, currently under construction, in which tanker robots recharge a population of worker robots to achieve continuous autonomous operation.

References

[1] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. A distributed memoryless point convergence algorithm for mobile robots with limited visibility. 15(5):818–828, 1999.

[2] J. Borenstein and Y. Koren. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, 1991.

[3] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, September 2002.

[4] J. Cortes, S. Martinez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. 51(8):1289–1298, 2006.

[5] G. Dudek and N. Roy. Multi-robot rendezvous in unknown environments, or, what to do when you're lost at the zoo. In *Proceedings of the AAAI National Conference Workshop on Online Search*, Providence, Rhode Island, July 1997.

[6] B. Gerkey, R. Vaughan, K. Stoy, A. Howard, G. Sukhatme, and M. Matarić. Most valuable player: A robot device server for distributed control, 2001.

[7] S. Gueron and R. Tessler. The fermat-steiner problem. *The American Mathematical Monthly*, 109(5):55–129, May 2002.

[8] A. M. J. Lin and B. D. O. Anderson. The multi-agent rendezvous problem. In *Proceedings of 42th IEEE Conf. Decision and Control*, Maui,Hawaii, 2003.

[9] Y. Kupitz and H. Martini. Geometric aspects of the generalized fermat-torricelli problem. *Bolyai Society Mathematical Studies*, 6:55–129, 1997.

[10] M. A. Lanthier, D. Nussbaum, and T.-J. Wang. Calculating the meeting point of scattered robots on weighted terrain surfaces. In *CATS '05: Proceedings of the 2005 Australasian symposium on Theory of computing*, pages 107–118, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.

[11] Y. Litus, R. T. Vaughan, and P. Zebrowski. The frugal feeding problem: Energy-efficient, multi-robot, multi-place rendezvous. In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2007.

[12] J. H. Mathews and K. K. Fink. *Numerical Methods Using Matlab*. Prentice-Hall Inc., 4 edition, 2004.

[13] G. Polya. *Mathematics and plausible reasoning, 2 vols*. Princeton, 2 edition, 1968. vol 1. Induction and analogy in mathematics; vol 2. Patterns of plausible inference.

[14] J. Rosen and G.-L. Xue. Computational comparison of two algorithms for the euclidean single facility location problem. *ORSA Journal on Computing*, 3(3), 1991.

[15] K. Schlude. From robotics to facility location: Contraction functions, 2003.

[16] S. Smith, M. Broucke, and B. Francis. Curve shortening and the rendezvous problem for mobile autonomous robots. 2005. Under revision.

[17] C. Wang. On the convergence and rate of convergence of an iterative algorithm for the plant location problem. *Qufu Shiyun Xuebao*, 2, 1975.

[18] P. Zebrowski and R. Vaughan. Recharging robot teams: A tanker approach. In *Proceedings of the International Conference on Advanced Robotics (ICAR)*, Seattle, Washington, July 2005.

[19] M. Zuluaga and R. Vaughan. Reducing spatial interference in robot teams by local-investment aggression. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta, August 2005.