# Multi-Robot Collaboration for Robust Exploration

Ioannis M. Rekleitis[1] Gregory Dudek[1]
Evangelos E. Milios[2]

[1]Centre for Intelligent Machines, McGill University,
3480 University St., Montreal, Québec, Canada H3A 2A7
[2]Department of Computer Science, York University,
North York, Ontario, Canada M3J 1P3

**Abstract**

*This paper presents a new sensing modality and stratagem for multirobot exploration. The approach is based on using pairs of robots that observe each other's behavior, acting in concert to reduce odometry errors. We assume the robots can both directly sense nearby obstacles and see each other. This allows the robots to obtain a map of higher accuracy than would be possible with robots acting independently bu reducing inaccuracies that occur over time from dead reckoning errors. Furthermore, by exploiting the ability of the robots to see each other, we can detect opaque obstacles in the environment independently of their surface reflectance properties. Two different algorithms, based on the size of the environment, are introduced, with a complexity analysis, and experimental results in simulation and with real robots.* [1]

## 1 Introduction

In this paper we discuss the benefits of cooperative localization during the exploration of a large environment. A new sensing strategy is used in order to improve the accuracy of the position estimation of each robot and hence the accuracy of the ensuing map. The robots explore the environment in teams of two; each robot is equipped with a *robot tracker* sensor that observes the other robot and reports its relative pose. The observing robot uses the position of its partner in order to update its own position estimate. Our approach is sufficiently robust to be able to cope with environments that may have uneven or slippery terrains, or whose surface reflectance properties are not well suited to conventional sensors.

Observe that conventional approaches to robotic mapping and navigation typically assume environments of rather limited size. Most existing approaches that function with real robots neglect issues like optimality and computational complexity. Further, the sensing techniques used to both explore the environment and position the robot often make rather optimistic assumptions about the environment: diffuse visual reflectors, substantial reflectivity, etc. In practice, surfaces may either be specular reflectors (mirror-like) or be hard to detect due to low reflectance. Furthermore real terrains may have frictional properties that make large-scale odometry difficult.

We deal with these issues in two ways, based on a polygonal approximation to the environment and the detection of convex (reflex) vertices. First, the two robots always move in such a way that they can see each other. More precisely, while one robot stays still the other robot moves, hence mapping the area swept by the line connecting the two robots as an area of free space. If an obstacle is located between the two robots, they can not see each other, thus detecting the obstacle. Second, the moving robot localizes itself with respect to the stationary robot thus improving its pose estimate independently from the conditions in the environment. The presence of reflex vertices is critical since it is these reflex vertices that determine the occlusion of regions of the environment. We use a pair of robots observing each other to build a map and circumvent problems of object visibility. The exploration strategy depends on the scale of the environment. When areas of free space are larger than the range of the robot tracker then a trapezoidal decomposition is used in order to guide the exploration. If the environment is small enough that it can be covered by the robot tracker then a triangulation of the environment is used.

The paper is structured as follows. In Section 2 we present the fundamental ideas in our approach of cooperative localization. In Section 3 we present an outline of the exploration algorithm used for mapping environments with large areas of free space. Section 4 covers the complexity analysis of the exploration, both the mechanical complexity and the computational complexity are examined. In Section 5 we examine experimen-

[1]Appeared in "International Conference in Robotics and Automation 2000", pages 3164-3169, San Francisco, USA, April 2000.

tal results from simulation and from laboratory experiments. Section 6 deals with extensions to more than two robots. Finally, Section 7 presents our conclusions.

## 2 Cooperative Localization

There are three potential sources of information for the localization of the moving robot. Odometry measurements $\hat{X}_{odom}(t)$ provide a base estimate of the moving robot's position (with high uncertainty $\sigma_o$). The different objects in the environment, when sensed from different positions, can provide updates in the robot position [2, 6]. Finally, the robot tracker $\hat{X}_{track}(t)$ provides measurements relative to the position of the stationary robot $\hat{X}_{stat}(t)$. A prime advantage of the tracker is that it is immune to variations in both the appearance and layout of the environment (e.g. due to moving objects) and is influenced only by the uncertainty in the position of the stationary robot $\sigma_s$ plus the error of the tracking subsystem $\hat{X}_{track}(t)$. The accumulation of uncertainty in the position of the stationary robot depends only on the number of role exchanges the two robots had. Consequently, over large open spaces where the odometry error grows without bound the moving robot could always refer back to a stationary landmark (a role that can be played by the second robot).

$$\hat{X}(t) = \frac{\sigma_s(\hat{X}_{track}(t) + \hat{X}_{stat}(t))}{\sigma_s + \sigma_o} + \frac{\sigma_o \hat{X}_{odom}(t)}{\sigma_s + \sigma_o} \qquad (1)$$

### 2.1 Tracker implementation

In this paper we will consider only an implementation of the robot tracker is based on visual observation of a geometric target on the robot [1]. Alternative possible implementations use retroreflectors or laser light striping – our actual robots are also equipped with such alternative technologies.

Each robot is equipped with a camera that allows it to observe its partner. The robots are both marked with a special pattern for pose estimation. The first part of the pattern is a series of horizontal circles (in fact these are cylinders and they project into almost linear patterns in the image). This allows the robot to be easily discriminated from background objects: the ratio of spacing between the circles is extremely unlikely to occur in the background by chance. Thus, the presence of the robot is established by a set of lines (curves) with the appropriate length-to-width ratio, and the appropriate inter-line ratios. The second component of the pattern is a helix that wraps once around the robot. The elevation of the centre of the helix allows the relative orientation of the robot to be inferred (see Figure 1). In practice, this allows the robot's pose to be

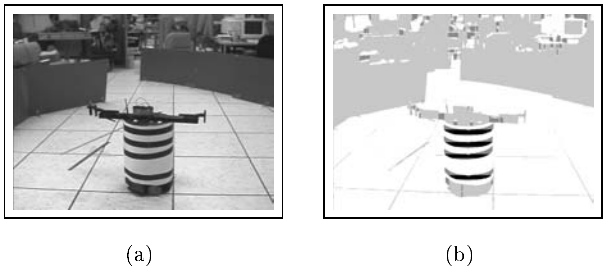
Figure 1: Robot Tracker: (a) The raw image of the moving robot as observed by the robot tracker. (b) The helical and cylindrical pattern detected in the image.

inferred with an accuracy of a few centimeters and 3 to 5 degrees.

By mounting the observing camera above (or below) the striped pattern, the distance from one robot to the other can be inferred from the height of the stripe pattern in the image, due to perspective projection (scaling of the pattern could also be used). The difference in height between the observing camera and the target can be selected to provide the desired tradeoff between range of operation and accuracy. One advantage of the helical target for orientation estimation is that it functions correctly even at very large distances (although with reduced accuracy, of course).

**Tracker calibration** We calibrate the camera system empirically using a lookup table to preclude nonlinearities in both the optical and computational subsystems. It is possible to estimate distances between roughly 180 and 450 cm with the camera configuration used in this experiment with a nominal accuracy of 1.5cm/pixel and 1.3°, although accuracy degrades with distance.

### 2.2 Exploration with the Robot Tracker

The two robots maintain an uninterrupted line of visual contact between them. When the moving robot proceeds along a trajectory the line of visual contact sweeps a wedge defined by the lines connecting the stationary robot position to the initial and final positions of the moving robots (see Figure 2) and the trajectory of the moving robot. If an obstacle obstructs the line of visual contact the moving robot backtracks and then proceeds to map around the interfering obstacle.

## 3 Exploration of large areas

In [4] we introduced an algorithm for exploring an area of size much larger than the sensing range of the robots. In environments consisting of large areas of open space (eg. warehouses, docking areas, open fields) it is quite common for the robots to be unable to follow a wall

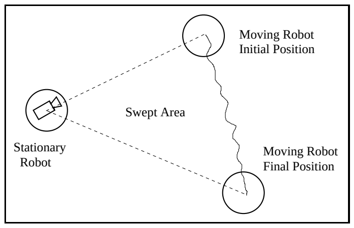

Figure 2: Area covered when one robot moves and the other one is stationary.

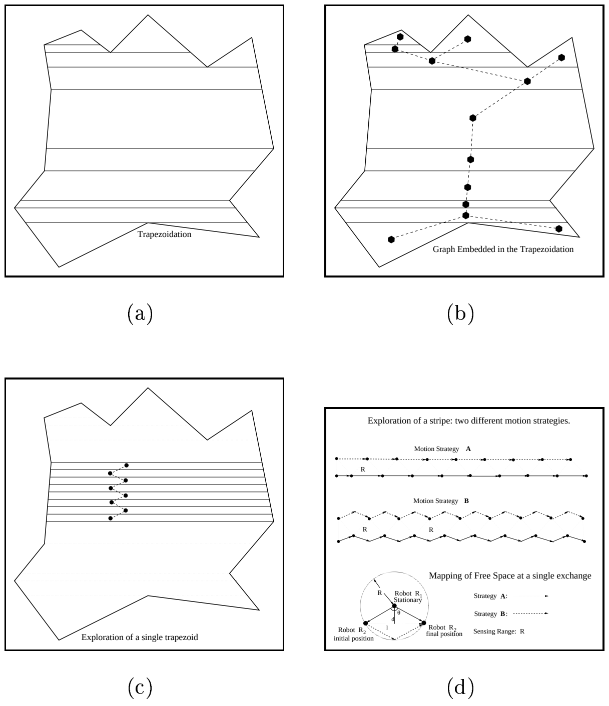
Trapezoidation
Graph Embedded in the Trapezoidation
Exploration of a single trapezoid
Exploration of a stripe: two different motion strategies.
Motion Strategy A
R
Motion Strategy B
R
R
Mapping of Free Space at a single exchange
R
Robot $R_1$ Stationary
θ
d
l
Robot $R_2$ initial position
Robot $R_2$ final position
Strategy A:
Strategy B:
Sensing Range: R
(a)
(b)
(c)
(d)

Figure 3: A top down description of the Trapezoidation algorithm. (a) The environment is divided into trapezoids. (b) The order of mapping given by a traversal of the *Dual Graph*. (c) Each trapezoid divided in stripes with a width proportional to the sensing range $R$. (d) Each stripe covered by areas of free space. Each area of free space is explored by the motion of a single robot.

or to detect any landmarks. In such environments the moving robot is using the stationary robot as a portable marker for relocalizing and mapping. Different motion strategies are examined for the complete mapping of the environment. The core idea of the algorithm is the mapping of an area of free space by one moving robot while the other robot is stationary. The purpose of the algorithm described below is to provide the order in which the free areas are going to be explored without repeating parts of the exploration and ensuring full coverage of free space. In a bottom up description of the algorithm there are the following operations. One robot moves and sweeps the line of visual contact across the free space, thus mapping a single region of free space. Then the two robots exchange roles in order to explore a chain of free-space areas which forms a stripe; a series of stripes are connected together to form a trapezoid. Finally, the entire collection of the trapezoids provides the trapezoidal decomposition of the entire free space – a complete spatial decomposition of the interior of the environment.

## 3.1 Outline of the Algorithm

The proposed algorithm is based on the trapezoid spatial decomposition of a polygon [3]. A top down description of this algorithm is illustrated in Figure 3a-d. More specifically, the two robots explore the world using a trapezoid decomposition of the free space as their guide, as can be seen in Figure 3a. Each trapezoid is mapped completely before the two robots proceed to the next one. The order in which the trapezoids are mapped is given by a depth first traversal of the embedded graph (see Figure 3b). Every trapezoid corresponds to a vertex in the graph; vertices corresponding to adjacent trapezoids are connected with an edge in the graph. The sensing range of the robot tracker provides a limit on the space that can be explored at any single time. Consequently if a trapezoid is larger than the range of the robot tracker then it is broken down into stripes with a width that depends on the sensing range **R** (see Figure 3c).

The exploration of a single stripe can be accomplished using different motion strategies. At the top of Figure 3d, two different motion strategies are displayed. One obvious approach (Strategy **A**) is, in each exchange, for one of the robots to move on a straight line (dotted line in figure 3d) sweeping (and hence mapping) a triangular region. The optimal strategy (Strategy **B**) is, in each exchange, for one of the robots to traverse the two chords shown as dashed lines in figure 3d), sweeping a diamond shaped area.

Strategy **A** is simpler, and requires a smaller number of changes in direction, but unfortunately, the width of the stripe ($d$) produced is suboptimal ($d < R$), and thus a larger number of stripes is needed in order to cover the same area. Strategy **B** is optimal in terms of path traveled over area covered (see Appendix A) because at any single time the width of the stripe covered ($d$) is the maximum possible ($d = R$). At the bottom of Figure 3d, the mapping of free space is presented over a single exchange. Angle $\theta$ is an input parameter, that can be chosen to minimize a cost as a function of $\theta$. In the case of reflex corners one trapezoid splits into two new trapezoids, and the two agents decide which branch of the embedded graph to follow.

When a series of explored regions are linked to each other as the exploration progresses, different types of stripes are created. In the case of the coverage of a triangular area, the two robots travel in parallel lines separated by $d$, and the stripe mapped has the same width $d$. In the case where each robot covers a diamond area, the trajectory of each robot would be a zig-zag line creating a stripe with width $R$ (equal to the sensing range of the robot tracker). A series of stripes connected together (lengthwise) map a single trapezoid.

## 4 Complexity Analysis

In order to analyze the complexity of the exploration we need to distinguish between two qualitatively different stages of exploration, the *local* and the *global* exploration phases. The local exploration strategy guides the path traveled for the mapping of a convex area of free space (a triangle, or a trapezoid). The global exploration strategy provides the order in which these areas are explored.

### 4.1 Complexity of Global Exploration

As noted earlier, the exploration strategy is guided by the dual graph of the spatial decomposition used. More specifically, during the trapezoidation algorithm the two robots explore one trapezoid at a time and then proceed to map the next trapezoid by following the dual graph in a depth first traversal. Every trapezoid is "*traversed*" twice, a first time when is being mapped and a second time when the two robots pass through in a shortest path traversal to visit the rest of the graph. When the triangulation algorithm is used, the dual graph is attached to the triangles. The two robots visit every triangle in a depth first traversal, thus passing through at most twice (the first time exploring, the second moving through towards the unmapped parts of the environment).

### 4.2 Complexity of the exploration over a single exchange

In contrast, when the trapezoidation algorithm is used, the moving robot could move through different trajectories as long as it stays inside the sensing range of the stationary robots. Different motion strategies present certain advantages and disadvantages. More precisely, there are different factors that affect the cost of the exploration depending on the configuration of the different robots. Every time the two robots exchange roles, the moving robot uses the stationary one to correct its position and then the stationary one starts exploring. Each of these operations introduces a time delay, therefore the number of exchanges increases the cost. In addition, every time one of the robots has to change directions the rotation adds to the total cost. Finally,

| Covering | Triangle Area | Diamond Area |
|---|---|---|
| Path length $P_\theta$ | $2Y + \frac{XY}{R}\frac{2}{\cos\frac{\theta}{2}}$ | $2Y + \frac{XY}{R}\frac{2}{\cos\frac{\theta}{4}}$ |
| # of steps $E_\theta$ | $\frac{XY}{R^2}\frac{2}{\sin\theta}$ | $\frac{XY}{R^2}\frac{1}{\sin\frac{\theta}{2}}$ |
| # of turns $R_\theta$ | $2\frac{Y}{R\cos\frac{\theta}{2}}$ | $2\frac{Y}{R} + 2\frac{XY}{R^2\sin\frac{\theta}{2}}$ |

Table 1: Analytical complexity of two different path curves.

the total path traveled has to be taken into consideration. For the two different motion strategies (diamond area covered, and triangular area covered) examined earlier, the total mechanical effort can be computed as shown in Table 1. The cost is calculated for the exploration of a rectangle $X$ by $Y$, when the robot tracker sensor range is $R$.

#### 4.2.1 Cost Analysis

The factors that affect the cost of the exploration are: the number of exchanges, the total path traveled and the number of rotations. For a specific team of robots the cost of the above factors could be determined beforehand. Specifically, the total cost of the exploration could be computed as the weighted sum of: the total path traveled ($P_\theta$) multiplied by the cost of path traveled ($C_p$ in sec/m), the total number of exchanges ($E_\theta$) multiplied by the cost for an exchange ($C_e$ in sec/exchange), and the total number of rotations ($R_\theta$) multiplied by the cost of rotation ($C_r$ in sec/rotation). The factors $(C_p, C_e, C_r)$ could be determined before the exploration, while the sensing range ($R$) of the robot tracker is known. Equations 2 and 3 provide the total cost $C_{total}(\theta)$ as a function of angle $\theta$ for the exploration, using diamond area and triangular area covering respectively, of a rectangle $X \times Y$ as a function of $\theta$, using the cost estimates and the analytical results from table 1. The optimal $\theta$ for the exploration is the one that minimizes $C_{total}(\theta)$.

$$\begin{aligned} C_{total,diamond}(\theta) &= C_pP_\theta + C_eE_\theta + C_rR_\theta = \\ &C_p(2Y + \frac{2XY}{R\cos\frac{\theta}{4}}) + (\frac{C_eXY}{R^2\sin\frac{\theta}{2}}) + C_r\theta(\frac{2Y}{R} + \frac{2XY}{R^2\sin\frac{\theta}{2}}) \end{aligned} \tag{2}$$

$$\begin{aligned} C_{total,triangle}(\theta) &= C_pP_\theta + C_eE_\theta + C_rR_\theta = \\ &C_p(2Y + \frac{2XY}{R\cos\frac{\theta}{2}}) + C_e(\frac{2XY}{R^2\sin\theta}) + C_r\theta(\frac{2Y}{R\cos\frac{\theta}{2}}) \end{aligned} \tag{3}$$

For one of our robots, a Nomad 200, the cost factors, for a typical experimental arrangement are: $C_p = 4.1$ sec/m, $C_e = 7$ sec/exchange, $C_r = 4.65$ sec/rad, the optimal angle $\theta$ is 180°, for the diamond area motion strategy. For the same costs the optimal angle $\theta$ is 90° for the triangular area motion strategy. As expected the total cost is lower for the motion strategy that covers the diamond area than that which covers a single triangular area.

## 5 Experimental Results

Experiments were conducted in simulation, using the robotic simulation package *RoboDaemon* [2] and in the lab. Experiments in the lab were used in order to validate the improvement in the localization based on the robot tracker over a pure odometry approach. In simulation a variety of odometry error models were applied in order to simulate different surfaces as well as different model worlds.

A pattern similar to the one traveled by the two robots was traced by one robot while the other one was stationary.

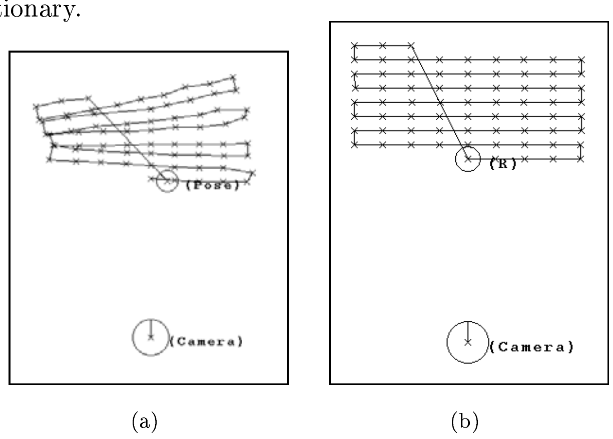

Figure 4: (a) Path of the moving robot estimated by the visual tracker. Precision is roughly 2cm. (b) Desired path of the moving robot. Although the robot can be driven along this path using open-loop control, dead reckoning errors lead to a substantial discrepancy.

In order to measure the accuracy of the map, a few locations along the path were selected and the position of the robot was estimated relative to the stationary camera. The accuracy of the positions estimated by the camera-based tracker was between 1.0 and 2.3 cm. As can be seen in Figure 4a,b the inaccuracy is largely due to rotational error and thus it is more evident near the sides of the rectangle. Figure 5 presents the absolute odometry error as it accumulates over the distance traveled by the robot.

## 6 More than two robots

The above strategies could be extended by the addition of more robots. For a team of $n$ robots the results

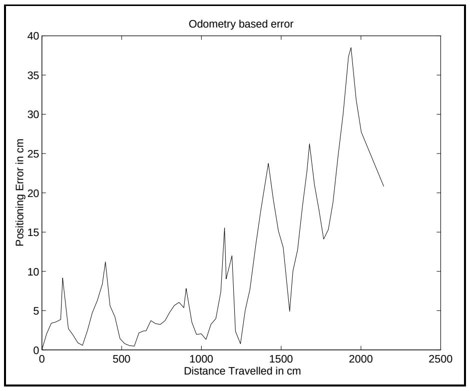

Figure 5: The error in positioning from the odometry estimations.

show longer range and more precision. By forming a chain of robots that “sweeps” through the free space the range of the tracker is multiplied by the number of robots, thus covering a much larger area in a single sweep. In addition, every robot could refer to more than one stationary robots, therefore gaining higher precision in its measurements. Two motion strategies are proposed with respective advantages. Using the first motion strategy, during the exploration only one robot moves while the stationary ones that are still visible are used as landmarks. This method ensures minimum uncertainty build up as, at any given time, the moving robot would correct its odometry error with respect to more than one landmarks. Using the second motion strategy, the robots divide into two teams and they interchange roles: while the first team is moving the second team works as a set of landmarks. This method explores an environment in less time but less robots are available as landmarks.

As mentioned earlier, an immediate extension of the trapezoidation algorithm can be obtained by the addition of more robots. When the two robots sweep one stripe of width $d$, by adding an extra robot (50% increase) we could double the area swept. In the original algorithm, every robot has only one device to track the other robots; in this case a scheduling algorithm should be applied in the order the robots are moving. If we add a second tracking device, one robot could track robots on both sides, allowing a parallel cover of double the area at the same time.

In the example in Figure 6 we use five robots ($R_0 \dots R_4$) that are positioned in two lines at time $T_0$. First the robots $R_0, R_2, R_4$ move forward, tracked by $R_1$ and $R_3$ accordingly, mapping the four triangles as free space(time $T_1$); then both $R_0, R_2$ track $R_1$, which moves forward (time $T_2$), while $R_2, R_4$ track $R_3$, which moves forward (time $T_2$). Then it is time for the other column of robots ($R_0, R_2, R_4$) to advance marking more

[2]RoboDaemon is robot control software employed at McGill University. It allows us to control the robots in the lab and also to perform experiments with simulated robots. The main advantage is that the simulated could be replaced by the real ones with no overhead.

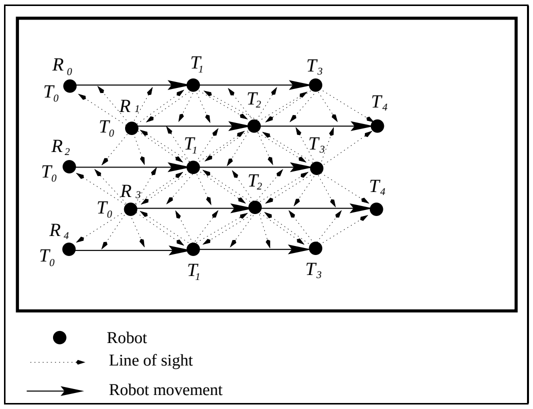

Figure 6: Exploration of a stripe with 5 robots.

area as free space (time $T_3$). The tracking is marked with the dotted lines of sight. The same pattern is followed as the two columns alternatively advance, marking a stripe of free space much wider than that possible with only two robots.

The second part of the algorithm concerning the exploration strategy for the whole space and the order in which the trapezoids should be explored is identical to the previous algorithm where only two robots were used.[3]

Moving only one robot at a time can also be easily extended to multiple robots. The robots start exploration aligned with each other in a straight line, at distance $R$ from each other, where $R$ is the tracker sensor range. This first robot and the last robot in the line act out the algorithm for two robots, while the role of the other robots is simply to provide a communication path between them. As such, the first robot in the line remains stationary, and the rest of the robots are moving such that the distance between two robots is never more than $R$. The width of the explored stripe is $(n-1)R$, where $n$ is the number of robots. A pictorial representation of this strategy can be seen in Figure 7.

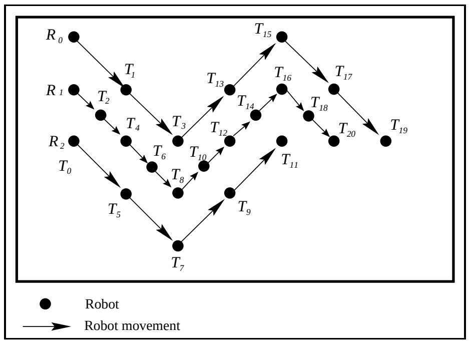

Figure 7: Exploration of a stripe with 3 robots, covering space in diamond areas.

[3]There is a possible speedup by splitting up the group in order to explore different parts in critical points, but that would in the end spread the robots too thin.

## 7 Conclusions

In this paper, we have described an approach to exploring and navigating in *large scale spaces* where positioning and obstacle detection might be difficult using traditional methods. In fact, such difficulties are likely to arise in many real-world environments.

Our approach is based on exploiting a line-of-sight constraint between two robots to achieve exploration with reduced odometric error. This approach can also cope with obstacles with hard-to-sense reflectance characteristics. Different algorithms were proposed depending on the scale of the environment. When the environment is small enough so that the robots can see each other from any two points on its boundary that have clear line of sight between them (i.e. they are never unable to see one another simply because they are too far away), then the triangulation algorithm is applied. If the environment is larger than the range of the robot tracker sensor then the trapezoidation algorithm is used. An open issue is how to automatically detect such situations *efficiently* during exploration and switch strategies, or switch back-and-forth between strategies based on local properties of the environment.

We are currently planning large-scale experiments of this strategy in a real physical environment. One difficulty that we must address is how to obtain accurate ground-truth to validate the performance of our approach over a large terrain. A standard practice is to simply observe the consistency and clarity of the resulting map and use this as a performance metric [6].

We are also considering combining this approach with more traditional localization methods (such as landmarks [5]) where they can be used effectively.

## References

[1] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Experiments in sensing and communication for robot convoy navigation. In *Int. Conf. on Intelligent Robots & Systems (IROS)*, v. 2, pp. 268–273, Aug. 1995.

[2] Paul MacKenzie and Gregory Dudek. Precise positioning using model-based maps. In *Proc. of the Int. Conf. on Robotics and Automation*, San Diego, CA, 1994.

[3] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994. ISBN 0-521-44592-2/Pb.

[4] I. Rekleitis, G. Dudek, and E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Int. Joint Conf. in Art. Intelligence* , v. 2, p. 1340–1345, Japan, Aug. 1997.

[5] Robert Sim and Gregory Dudek. Learning visual landmarks for pose estimation. In *Proc. of Int. Conf. on Robotics & Automation (ICRA)*, May 1999.

[6] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Mach. Learning & Autonomous Robots*, 31,5:29–53,253–271, 1998.