Reducing odometry error through cooperating robots during the exploration of an unknown world.

I. REKLEITIS[1] G. DUDEK[1] 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

We consider how to cover and map an initially unknown environment using two (or more) mobile robots. Most mobile robot systems accrue odometry error while moving, and hence need to use external sensors to recalibrate their position on an ongoing basis. Unfortunately, most sensing systems are constrained with respect to the types of environment in which they are suitable. We deal with position calibration and odometry error by using multiple robots for exploration. This allows them to use one another as landmarks. We consider how exploration can be efficiently accomplished and how a large environment can be divided and conquered.[1]

1 Introduction

Several interesting potential applications of robotics technology would benefit from the use of mobile robotic systems. This, in turn, suggests the need for mobile robot systems that can explore an environment and automatically build an internal model of a map of where they can go and where objects are located. The applications for such systems range from hospital delivery scenarios (where the arrangement of obstacles may vary) to extra-terrestrial planetary exploration.

While mapping can be accomplished by a single robot, most mobile robot systems accrue odometry error while moving, which makes pure dead reckoning undesirable. This suggests the need to compute a robot's position using external landmarks or features on an ongoing basis. Unfortunately, most sensing systems are constrained with respect to the types of environment in which they are suitable. The selection of a best type of general purpose landmark and the selection of an associated sensor and algorithm is still an active research issue. We deal with position calibration and odometry error by using multiple robots for exploration. This allows them to use one another as landmarks and avoids the need for strong assumptions regarding the appearance of the environment. We consider how exploration can be efficiently accomplished and how a large environment can be divided and conquered. This paper deals primarily with questions of efficiency and feasibility from a theoretical standpoint, related issues, including more than two robots, are also addressed in [10]. We model the world as a collection of simple planar polygons with obstacles represented by holes.

In the case of an ideal robot with no odometry error and an ideal range scanning sensor, Lumelsky [6] was one of the first to develop provably correct exploration strategies, one of which is based on circumnavigating successive objects. This approach, like several others, assumes that there are no dead reckoning errors and that sensors return perfect data. Other techniques [9], representative of existing approaches, assume a polygonal world, which the robot maps by traversing the visibility graph ensuring every part is visited. Some models deal with the world at a purely topological level [3, 5]. Experimental approaches to environment exploration have also been developed [1, 2], demonstrating satisfactory performance in limited environments but without a performance guarantee. In contrast to these approaches, we present theoretical results, but deal explicitly with the need to compensate for odometric error, and consider sensing whose accuracy deteriorates with increasing distance. We compensate for these problems by using multiple cooperating robots to explore the environment.

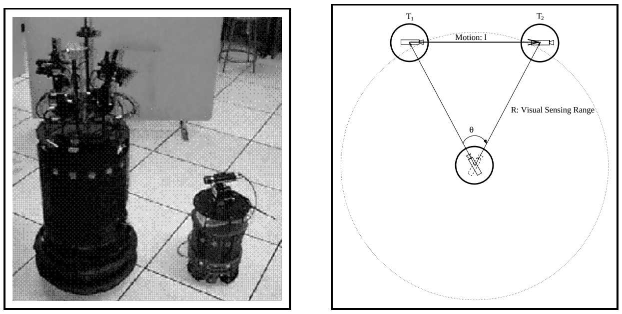
The organisation of this paper is as follows. In Section 2 we present the description of the world and the robot model. In Section 2.1 we analyse the advantages of cooperative robots versus a single one. Then in Section 3 an algorithm for exploring large areas (compared to the sensing range of the two robots) is presented. In Section 4 a triangulation algorithm is analysed. Finally Section 5 contains conclusions and suggests possibilities for future work.

2 Problem definition

In this paper we deal with the exploration of a purely two-dimensional environment using a point robot. We represent the environment by a simple planar polygon with holes. A polygon is *simple* if there is no pair of non-consecutive edges sharing a point. The model of the world is essentially a set of simple polygonal obstacles contained within a larger polygonal boundary.

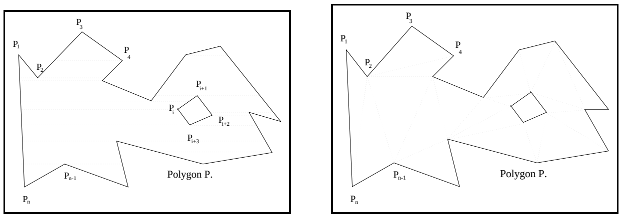
---

[1] Appeared in "Fifth IASTED International Conference ROBOTICS AND MANUFACTURING", Cancun Mexico, May 1997



(a) (b)

Figure 1: (a) Two robots equipped with Vision and Sonar sensors. (b) The model of the robot movement and the area covered.



(a) (b)

Figure 2: (a) Trapezoidation of a simple polygon with holes. (b) Triangulation of the same polygon.

Our model for robots is minimal but is easily generalised. For the complexity analysis, the robots are modelled as points that can move in any direction, and they are equipped with two sensors. In our laboratory, the experimental setup includes two different robots (see Figure 1a). The one on the left is a Nomad 200, while the one on the right is a RWI B-12. Both are equipped with two different kinds of sensing systems. The first sensor is an *object detector*, able to detect any object in the immediate vicinity of the robot. The object detector allows wall following and object avoidance. It is implemented by a sonar ring but could also be based on an infra-red device or even a tactile sensor. The range of the object detector is limited. The second sensor is a *robot tracker*, with the ability to locate another robot when there is a free line of sight between them, and determine the distance to the second robot as well as its orientation. Examples of this type of sensor are a vision system that could locate a pattern on the other robot or a laser range finder and a retroreflective target on the other robot. In prior work, we have described the use of a simple but robust vision system for the robot tracker [4]. We assume that the range of the robot tracker is much larger than that of the object detector (i.e., we can see further than we can reach). Both robots in our laboratory are equipped with a sonar ring that allows them to detect any object that is close to them, and a camera mounted on a pan-tilt unit that allows the stationary robot to track the position of the moving robot, sweep the free space, and accurately report back the position of the moving robot in order to correct any positioning errors (see Figure 1b).

The robots explore the unknown environment by progressively covering free space in the polygonal world. Several planar decompositions have been proposed in the computational geometry literature [7]. Although they apply to worlds that are completely known, they can be used as a starting point to develop "on-line" versions that construct the decomposition as part of the exploration process. The advantage offered by this approach is guarantee of full coverage without duplication and a standard description for use in higher level reasoning.

Two systematic methods can be seen in Figure 2. The first method is to cover free space with trapezoids, as in Figure 2a. The second method is the decomposition of a simple polygon by triangulation. The interior of the polygon is decomposed into triangles without adding vertices by using non-intersecting diagonals (see Figure 2b)[8].

2.1 Advantages of multiple robots

The design of a robust, error-free, general-purpose range sensor has remained a difficult challenge. In general, high accuracy entails a limited range of operation for most devices. It is possible in many applications to consider the robot and its sensor range as a *point* or a small disk that covers the space by moving through it. In such a case the overall path necessary to be travelled before the whole map is constructed defines an area-filling curve swept by the robot/sensor system.

On the other hand, having one robot of a two-robot team observe and track another cooperating robot is a comparatively simple task (since there is not need to measure reflected energy from unpredictable materials in the environment, as is the case with a range sensor). If we use a pair of robots with the above described tracking sensors, then by moving one of them across the base of a triangle (for example AB) with the other at the opposite corner (for example C) they would cover the area of the triangle ($\frac{1}{2}|AB|\alpha$ where α the distance of C to AB), by travelling only the distance $d = |AB|$. This can constitute an arbitrarily large improvement over a space-filling sweep algorithm[2].

Another problem that arises in practice is odometry error. Due to imperfections in the construction of a real robot and the properties of the environment, mobile robots cannot avoid building up small errors in their position and orientation estimates when they move. After several steps, the robot's estimate of its position can be very different from the actual position. The traditional self-contained solution for the localisation problem is to correct the robot's position estimate by making reference to external landmarks observed using the robot's

---

[2] In practice, even line of sight tracking is range limited and can be described as a sweep, but in this case the sweeping figure can be extremely large.

sensors.

In our work, two or more robots are used in conjunction to limit the size of odometry errors. This is accomplished by having only one robot move at any time, while the other robot(s) observe it. This allows them to track it and measure its position. Later on, the roles are reversed: the robot that had been moving becomes the observer while another robot can move. For now, full communication is assumed, as the moving robot can obtain its current position from the observer's position at any time [4]. This allows positioning to be accomplished based on the observing robot's positions and independent of any environmental characteristics.

The exploration strategy proceeds by dividing space into regions and by sweeping out each region with the line segment defined by the line-of-sight between the robots. We represent each of these regions by the vertices of a graph, and regions that are adjacent are connected by an edge. A "high-level" description of our exploration strategy can then be formulated in terms of the exploration of this graph.

3 Decomposing space into trapezoids

The stripe-like decomposition described above is now used as a starting point for developing an exploration algorithm with two robots. The two robots are modelled as points, and are each equipped with an object detector and a robot tracker that can accurately locate the other robot up to a distance R (see Figure 1b).

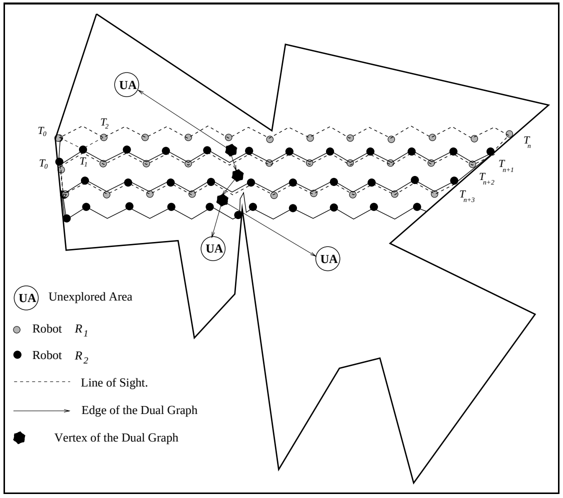


Figure 3: Application of the trapezoid algorithm with two robots.

The exploration algorithm consists of two logical parts: 1) *local* exploration that sweeps a horizontal stripe of free space inside one trapezoid, 2) *global* exploration that connects the stripes together and decides which part to explore next.

The two robots are "awakened" at time T_0 next to each other. The robot R_2 moves away from robot R_1, which remains stationary until R_2 reaches a distance $d = R$, distance that gives the maximum covered area, while accurately locating the position of R_2, (time T_0 in Figure 3a). Consequently, robot R_2 moves to a new position T_1, then they switch roles and robot R_1 travels to position T_2 (see Figure 3a).

The two robots switch roles as they travel across the environment mapping a stripe of free space until they reach a wall (positions T_n and T_{n+1} in Figure 3a,) then they reposition themselves to explore the next stripe below (positions $T_n + 2$ and T_{n+3}). The two robots move from left to right during the exploration of the first stripe and then change direction every time they explore a new stripe. The order in which the stripes are explored is given by traversing the dual graph. The dual graph is constructed by matching every stripe to a vertex of the graph, and for every pair of adjacent stripes an edge connecting the corresponding vertices is added. When the two robots encounter a reflex vertex[3], two new edges are introduced to the dual graph, and they choose one of the two edges to continue the exploration, while making a mark on the map for the entrance to the unexplored territory.

After an area is completely mapped, the two robots locate the nearest unexplored territory, plan a safe path, and move there. Although the movement through already-mapped free space is safe, extra care is taken in order to avoid odometry error, therefore only one robot moves at a time and the second remains stationary and reports their location.

In our approach a depth first search strategy is used in order to determine which edge of the dual graph the robots are going to follow in the exploration. It is worth noting that in order to have optimal results the deepest branch of the graph should be explored last, but without a-priori knowledge this can not be determined in advance. The trapezoid decomposition covers the entire free space with a finite number of stripes (trapezoids). The algorithm methodically explores every one of these stripes, and it never repeats the exploration.

4 Progressive Triangulation

This algorithm operates in an environment where the visual sensing range is at least as large as the diameter of the polygon. The output of this algorithm is a map of the free space decomposed into triangles. If the stationary robot is placed at one corner of the polygon and the other sweeps through an opposite edge, then the defined triangle is part of a triangulation of the polygon.

In traditional approaches to this kind of problem in the context of computational geometry, the location and the ordering of the vertices is considered known. In the exploration context, however, the vertices and edges are unknown and they are discovered incrementally as we map the area. Moreover, while the time cost for repeated visits to an edge in most existing non-robotic algorithms

---

[3] For a simple polygon P a reflex vertices are the concave vertices. For obstacles, inside the polygon, reflex vertices are the convex vertices of the obstacle

is of little consequence, in mobile robotics the cost greatly depends on the distance between the vertices.

We start from an arbitrary position of the environment and proceed to map it as a set of convex polygon/shapes of free space connected as a graph in the case of a simple polygon with holes, or as a tree in the case of a simple polygon. As an initial step, the two robots sense the closest wall, one of them (R_1) stays stationary and the other (R_2) travels until it is next to the wall, and then the roles are exchanged and R_1 moves also to the wall; then first R_2 and then R_1 move to the opposite corners of the approached edge. It is at that point that the triangulation algorithm starts. The robots alternate roles moving along the edges of the polygon mapping the free space inside it.

The application of the algorithm in the case of a simple polygon with with many reflex vertices is shown in Figure 4. The position of a robot is marked by the time it arrived there, for example, position T_i. The robots are positioned exactly on top of the edges/vertices in the figure while in reality they would keep a minimum distance from them. The two robots start at time T_0, at the two corners of an edge (both marked as T_0), and they proceed to map the complete polygon. They finish at time T_{12} with two unexplored areas left (marked with **UA**). Every time a triangle of free space is mapped one more node is added to the dual graph. Every time a reflex vertex is encounter, two extra edges are added to the dual graph, and then a decision is made which one to follow. In the example of Figure 4 this happens at T_5 and T_{11}.

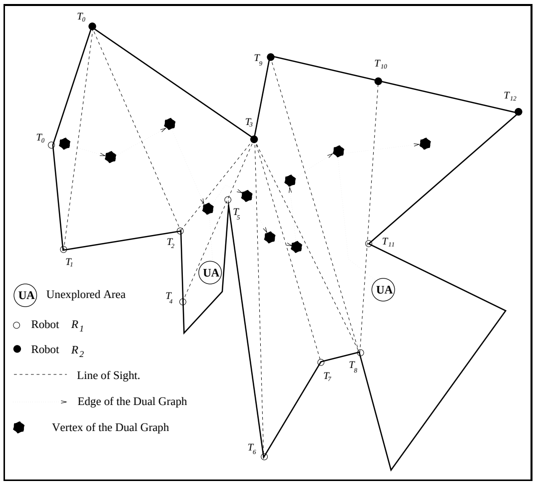


Figure 4: Triangulation like exploration of an unknown environment. The general algorithm.

The complexity of the triangulation algorithm could be defined as a function of the distance travelled by the two robots and broken down in two parts. The first part is the distance travelled during the exploration of the polygons in triangles, and the second is the travel that is necessary when they have to traverse back parts of the explored graph in order to reach the unmapped areas. The first part consists of the perimeter of the polygon (and the perimeter of the holes). The second part occurs when reflex vertices are visited, and is less than a distance nD where n is the number of reflex vertices and D is the maximum diagonal. The algorithm always terminates because the number of triangles explored is finite and there are no repetition on the exploration.

The two algorithms should be used together, when the robots approach a close space where they could *see* each other from wall to wall, the triangulation algorithm should be used to map it, when they move into an open area then the trapezoid decomposition algorithm should be used to sweep the area.

5 Conclusions

In this paper two new algorithms for exploring an unknown environment are proposed. Both algorithms use a well known planar decomposition form in order to systematically explore the free area of an unknown environment modelled as a simple polygon with holes. The trapezoid decomposition is used for large areas ensuring an exploration strategy that finishes with the total free space mapped as a set of trapezoids. For small areas a triangulation of the free space is returned.

Realistic assumptions, such as odometry error and sensing that deteriorates with distance, are made, and the advantages of cooperation are evident. Both algorithms return a complete map, while a single robot would encounter great difficulties in such a case.

References

[1] Tucker Balch and Ronald C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52, 1994.

[2] H. Bulata and M.Devy. Incremental construction of a landmark-based and topological model of indoor environments by a mobile robot. *Proc. IEEE Conference on Robotics and Automation*, pp 1054–1060, April 1996.

[3] Xiaotie Deng and Andranik Mirzaian. Competitive robot mapping with homogeneous markers. *IEEE Trans. on Robotics and Automation*, 12(4):532–542, August 1996.

[4] G.Dudek, M.Jenkin, E.Milios, and D.Wilkes. Experiments in sensing and communication for robot convoy navigation. In *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pp 268–273, Pittsburgh, PA, August 1995.

[5] B. Kuipers and T. Levitt. Navigation and mapping in large-scale space. *AI Magazine*, pp 25–43, summer 1988.

[6] Vladimir J. Lumelsky, Snehasis Mukhopadhyay, and Kang Sun. Dynamic path planning in sensor-based terrain acquisition. *IEEE Transactions on Robotics and Automation*, 6(4):462–472, August 1990.

[7] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford U. Press, NY, NY, 1987.

[8] J.O'Rourke *Comp. Geom. in C* Cambridge U. Press '94.

[9] N.S.V. Rao. Robot navigation in unknown generalized polygonal terrains using vision sensors. *IEEE Trans on Systems, Man, and Cybern.*, 25(6):947–962, Jun 1995.

[10] Ioannis Rekleitis, Gregory Dudek, and Evangelos Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Int. Joint Conf. on A.I. (see also CIM Tech. Report TR-CIM-97-01)*, Aug. 1997.