Frontier-Graph Exploration for Multi-robot Systems in an Unknown Indoor Environment

Mark Gossage, Ai Peng New and Chee Kong Cheng

Cooperative Systems and Machine Intelligence Lab, DSO National Laboratories, Singapore {gmark, naipeng, ccheekon}@dso.org.sg

We present a new method for single/multiple robot indoor exploration and mapping. The algorithm combines local Frontier-based exploration technique and global graph-based representation of the environment to produce a robust autonomous exploration strategy. This graph is used and shared to allow cooperative exploration. Our implementation is fully decentralised and has no central control to organise the robots, it is also robust to failures both in communications and robot attrition. Our approach has been demonstrated to work on a team of two Pioneer 3AT robots in an area of $50m^2$. In the simulator it has been successfully scaled to a team of five robots in a map of over a hundred rooms and an area of $5000m^2$.

1 Introduction

The aim of this research work is to develop a robust exploration technique for single and multi-robot system in an unknown indoor environment. For the case where multiple robots are employed, the system must be scalable, decentralised, tolerant to temporary lost of communication between some robots and able to handle robot attrition.

Our approach uses concepts from the Frontier-based exploration and graph-based representation of the environment. Instead of using a global occupancy grid (OG), our method uses a fixed size local OG centred on each robot. As the robot explores, the frontiers located in the OG are used as graph nodes on a global graph. The robot uses this global graph for high-level planning and navigation. For multi-robot system, the graph is regularly synchronised between robots to ensure a consistent map. Cooperative exploration is performed through tasks negotiated between robots.

Section 2 discusses other related work in single robot indoor exploration and multi-robot systems. Section 3 describes our Frontier-Graph Exploration algorithm (FGE) in detail. In section 4, the simulation and hardware testing results are presented and discussed. Finally in section 5, further work is considered.

2 Related Work

Exploration of an environment is a complicated problem for robots to solve. It consists of many sub-problems, many of which are non-trivial to solve. Problems such as Simultaneous Localisation and Mapping (SLAM), determining where to explore next, deciding how best to organise multiple platforms have all enjoyed substantial attention over the years. From our survey of exploration algorithms, there are only a few map representation methods. The majority are based upon grid based map representation (Burgard et al. 2002; Zlot et al. 2002, Simmons et al. 2000, Yamauchi 1998) with a few others looking at topological map representations (Choset and Burdick 1995a, 1995b).

By far the most common method for multi robot exploration is to use an occupancy grid (Moravec 1998) with a frontier based exploration algorithm (Yamauchi 1996). It is a simple and proven method of exploration and mapping. But the grid based map has a fundamental flaw associated with it, its size. Grid maps (especially those of high resolution) require large amounts of memory (usually in the order of megabytes). With the growth rate of computer memory, this does not necessarily present a problem. However in a multi-robot context, such maps must be shared regularly. Bandwidth consumption becomes very high as the entire map is needed for exploration and path planning.

The other method of map representation is topological (graph based) such as the Generalised Voronoi Graph (GVG) (Choset and Burdick 1995a, 1995b). Though the graph representation is simple and compact, the GVG is a technique primarily for exploring tunnel-like environments or corridors. In a cluttered environment (such as an office), it becomes difficult to create the graph, and the graph complexity increases greatly.

This paper presents a hybrid exploration and mapping algorithm, called ‘Frontier-Graph Exploration’ (FGE). It attempts to use the best parts of both grid based maps and topological maps while mitigating the various limitations of each representation.

The idea of developing a hybrid grid-topological representation is not new. Thrun and Bücken (Thrun and Bücken 1996) showed a method converting a grid based map into a region based (topological) map. More recently the Centibot project (Konolige et al. 2004) used such a map in its Spatial Reasoning component. After the initial mapping (using an occupancy grid), an offline process was applied to turn this into a Voronoi diagram, and to the final graph. The graph was then used by the robots for sharing and navigation about the map.

Our method distinguishes itself from these past works in that it is generated online as the robot explores, not as a post-processing step after the exploration is performed. It also does not require the global occupancy grid that the other methods require, but uses a local grid only. This substantially reduced bandwidth usage, as no grid information is ever shared, only the graph information. It is also a decentralized approach with robots individually creating and sharing graphs as they explore.

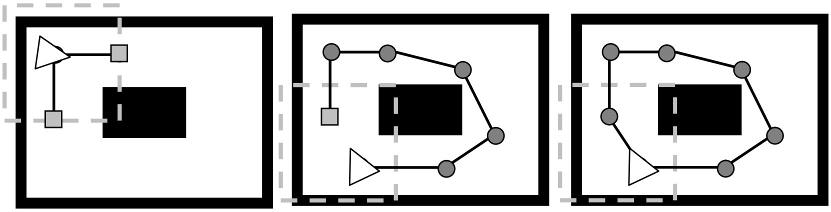
3 Frontier-Graph Exploration

3.1 Single Robot Frontier-Graph Exploration

The FGE algorithm assumes a 2D environment, a robot mounted with an accurate sensor (such as a Laser Range Finder), and good localisation.

The main perception component of the FGE is the 'Circular Perception'; a small fixed size local OG (Elfes 1990) formed by accumulating sensor data as the robot moves in the environment. It is used to determine obstacles, configuration space and frontiers for exploration. Unlike a normal occupancy grid, which will continually expand as the robot moves, the Circular Perception is a fixed size grid that scrolls about with the robot, always keeping the robot in the middle of the grid. This perception provides sufficient information for local sensing and navigation, but consumes a fixed amount of memory.

The basic FGE algorithm can be summarised as: find the frontiers on the local OG, add them to the graph, select a frontier (using a simple cost function) and move to it, repeat until no frontiers exist. A step by step diagram of the exploration can be seen in figs. 1a-c.

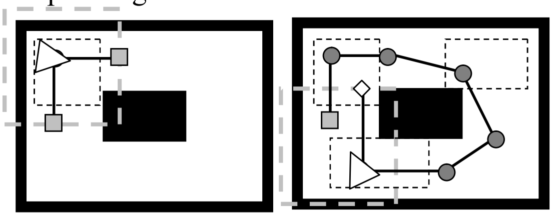


(a) (b) (c)

Fig. 1a-c. Step by step frontier-graph exploration. The triangle represents the robot, dashed lines are the circular perception, squares are unexplored nodes, circles are explored nodes

The first problem with this approach was how to determine when the robot moves into space that has already been explored. In Frontier exploration, this is taken care of by the global OG. But a local OG does not hold this history. Therefore when the robot returns to an explored space, it may add new frontiers in the explored area. For example in the fig. 1b, the robot has clearly completed the exploration. However without the global knowledge, it might continue adding new nodes to the graph and would continue to explore indefinitely.

In order to overcome this problem, the FGE algorithm stores and makes use of the clear space region information. The clear space region is defined as a polygon approximating the clear (non-configuration space) area that is around the robot. Each explored node in the graph includes the clear space region information.



(a) (b)

Fig. 2a-b. FGE with added region information (dotted line, not all regions are added for clarity). The diamond represents a possible frontier that will not be added into the graph as it falls within an existing region

In figs. 2a & b above, the robot has performed FGE and added region information into the graph. As it nears the completion of the exploration, it reaches back to an explored area (fig. 2b). The Circular Perception reports a frontier at the point marked with a diamond. However this point is within an existing clear space region (the region attached to the first node), therefore the frontier should not be added to the graph.

The second problem was that of creating too many frontier nodes. As the robot finds frontiers, they are added to the graph. In normal frontier exploration, as the robot moves to a frontier, other nearby frontiers will be removed. However within FGE, graph nodes must still be explored to ensure completeness. This causes the robot to perform useless work, exploring nodes that will not yield new information.

Using the region information, it is possible to identify these 'extra work' nodes. Upon arriving at a node, all the unexplored nodes, which are in the current clear space region, but not near to a frontier, can be considered 'unnecessary to explore'. In fig. 2b when the robot reaches the final node, the current region information, overlaps the last remaining unexplored node (the square). Since it is clear that the node is in the explored space and not near any frontiers, it can be marked as 'unnecessary to explore'.

To summarise, the full Frontier-Graph Exploration Algorithm is as follows:

1. Move forward a distance to perform initial population of circular perception.
2. Add a graph node (type: explored) at the robot's current location as the starting point, with the clear space region attached.
3. For each node within the current clear space region:
   - Add link between the node and the current node (unless present).
   - If the node is an unexplored node and is not within a certain distance of a frontier, mark the node as unnecessary to explore.
4. For each of the frontiers found within the circular perception:
   - If the frontier is within an existing region (excluding the current region) do not add this.
   - Else add this frontier as an unexplored node and add an edge linking the current node to this unexplored node.
5. If there are unexplored nodes, select an unexplored node (according to the cost function) and move the robot to the node. Otherwise end.
6. Upon arriving at the node, change that node's type from unexplored to explored.
7. Goto Step 3.

3.2 Multi-Robot Frontier-Graph Exploration

Building on the foundation of single robot FGE, we extended the algorithm to multiple robots. The two main features to enable multi-robot exploration are graph merging, to maintain a consistent global graph, and node ownership/negotiation mechanism.

In Multi-Robot FGE, we assumed the robots have good localisation and a common frame of reference, but not necessarily the same start position. We also assumed that both robots and communication could fail. Within the simulator, the localisation assumption is acceptable. Within the hardware however, this is not so easy to achieve.

In order for each robot to maintain a consistent global graph, the graph was regularly broadcast to all other robots. As the global graph is quite compact (even with the region information), it is feasible to send the entire graph in one go. Because of this, robots could maintain a consistent graph unless they were out of communication range for some time.

The individual robots do not attempt to maintain a common numbering scheme for its nodes and edges, but instead use the location of the nodes to merge the graph. This requires the common frame of reference, but is relatively simple to do. When a robot received a graph update, it would match each node in turn with its existing nodes using the nodes location (with a small distance tolerance), and then add any new nodes to its graph. At the same time, it would build a temporary mapping between the received graph node ids and its own graph node ids. Then the edges would be added, using the mapping to determine which nodes should be linked. In practice this was found be acceptable, and performed well in simulation and hardware, provided the localisation was consistent.

The node ownership/negotiation mechanism is used to help the robots coordinate among themselves. Each node is assigned an owner (the robot who discovers the node). When a robot needs a task to do, it takes all unexplored nodes in the graph and sorts them into a list using a cost function based upon the distance between the robot and the node (using Dijkstra's shortest path).

If the lowest cost node is owned by the robot, it performs exploration as normal. If it does not own the node, it must contact the owner and negotiate with them for node ownership. If a robot is unable to obtain the node, it will then consider its second choice and so on. Should no nodes be available, the robot will wait for new ones to appear.

When a robot is contacted with a negotiation request it will agree to transfer the node if the robot is not planning to do it. If it is moving to the node, then it would compare its distance to node with the requesting robot, and agree if the requester is nearer the node.

If the requesting robot could not contact the node owner, it would assume that the owner has either failed, or is too far away from the node to explore it. In which case, it would 'steal' node ownership. Though this assumption is not always correct, it allow the overall algorithm to be robust to communication/platform failures.

4 Results and Discussion

We performed both single robot and multiple robot exploration (only multi-robot results presented) in both the Player/Stage simulation and on our Pioneer 3AT hardware.

4.1 Multiple Robot Simulation

Multi-robot simulation was conducted in the hospital environment as shown in fig. 3. The hospital map is the largest map available with Player/Stage simulator. Measuring around 120m x 50m, and with over a hundred rooms, it is a formidable test for robotic exploration. A team of five simulated robots managed to explore the map in around forty five minutes, generating a graph of several hundred nodes in the process. During our tests, one of the robots failed. However the remaining robots were able to cope with this loss and complete the task without issue.

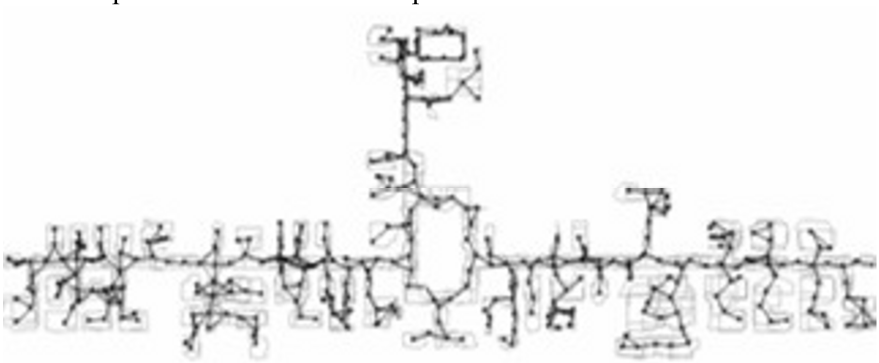
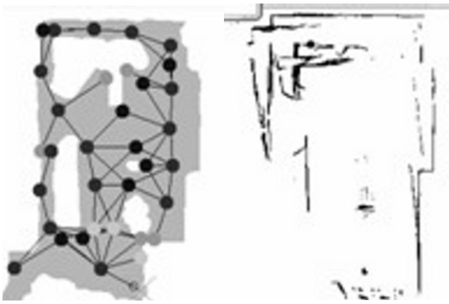


Fig. 3. Multi-Robot FGE in a simulated environment of 120m x 50m, showing the graph nodes and the regions.

4.2 Multiple Robot Hardware

The FGE algorithm was tested in hardware using two Pioneer 3AT's in our company's lobby area. It is consists of a relatively open area, and is about $50m^2$ in size.

One of the biggest issues during hardware testing was localisation. We used a decentralised EKF based SLAM algorithm to provide localisation. It was mentioned earlier that FGE required a common frame of reference for graph sharing and merging. However errors in the localisation often caused the robots graphs to become misaligned with each other. This would cause robots to be unable to navigate to the tasks that they negotiated from other robots.



(a) (b)

Fig. 4a-b. Multi-robot FGE in a $50m^2$ lobby area, showing the graph and regions (a), and the point cloud map (b)

Fig. 4a-b shows the results from two robots performing FGE. Looking at Fig. 4b, it is clear that some localisation drift has occurred which caused the echoing effect in the point cloud.

Attempts to expand beyond two robots proved impossible as the localisation errors became worse when tests were run with more robots. More work is needed in the localisation area to rectify this problem. However the result here serves to demonstrate that the FGE can be realised in the hardware.

4.3 Issues

Besides localisation, there are two other issues observed: inefficiency in the cooperation and occasional missed frontiers. Often robots stopped to wait for new node to appear or crowded together 'fighting' for nodes. This is generally caused by the simple greedy heuristic used. This could be overcome by using a better cooperation strategy.

The missing frontiers issue is shown in Fig. 5. Here a frontier is on the right of the perception, but due to the narrowness of the configuration space obstacle, the FGE is unable to place a node near to the frontier and it is missed. This limitation causes some frontiers which are beyond narrow openings to be missed. A possible solution to this might be to use some kind of local path planner, though how best to represent this on the graph is still not clear.

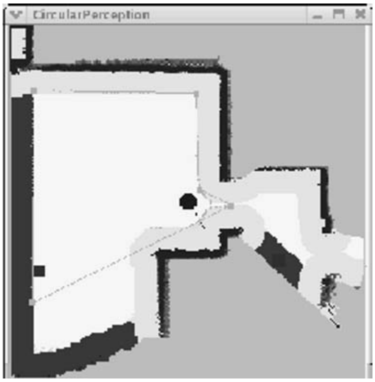


Fig. 5. Due to narrow opening to the right of the robot, it misses the frontier to the right of the robot

5 Further work

The FGE algorithm has several area's for future work. One area is to use an alternative method for task selection and cooperation methods, using ideas from some of earlier mentioned papers (Burgard et al 2002, Simmons et al 2000, Zlot et al 2002).

Localisation is a key area for improvement upon. Several authors (Lu and Milios 1998, Gutman and Konolige 1999) consider a map to be a graph like structure with laser range finder scans attached to each node. This is similar in structure to the FGE, though the purpose is for mapping/localisation. Producing an integrated exploration, mapping and localisation algorithm looks to being a very promising area forward.

The final area for exploration is that of node placement. The current FGE algorithm places its nodes at the limits of its perception, making it a non-repeatable method. An alternative would be to consider using a method closer to the Voronoi Diagram, such as in (Konolige et al. 2004). This would provide a more repeatable method for placement of nodes. A comparison of these methods is an area for future research.

References

Burgard W, Moors M, Schneider F (2002) Collaborative Exploration of Unknown Environment with Teams of Mobile Robots. Lecture notes in Computer Science, volume 2466 pp 52-70

Choset H, Burdick J (1995) Sensor based planning, Part I: The Generalized Voronoi Graph. In Proc. IEEE Int. Conf. on Robotics and Automation

Choset H, Burdick J (1995) Sensor based planning, Part II: Incremental Construction of the Generalized Voronoi Graph. In Proc. IEEE Int. Conf. on Robotics and Automation

Elfes A (1990) Occupancy grids: A stochastic spatial representation for active robot perception. In Proceedings of the Sixth Conference on Uncertainty in AI. Morgan Kaufmann Publishers inc.

Gutman J, Konolige K (1999) Incremental Mapping of Large Cyclic Environments. Proceeding of Conference on Intelligent Robots and Applications

Konolige K, Fox D, Ortiz C, Agno A, Eriksen M, Limketkai B, Ko J, Morisset B, Schulz D, Stewart B, Vincent R (2004) Centibots: very large scale distributed robotic teams. In Proceedings of the International Symposium on Experimental Robotics

Lu F, Milios E (1998) Robot pose estimation in unknown environments by matching 2d range scans. Journal of Intelligent and Robotic Systems

Moravec H (1998) Sensor fusion in certainty grids for mobile robots. AI magazine, pp 61-74

Simmons R, Apfelbaum D, Burgard W, Fox D, Thrun S, Younes H (2000) Coordination for multi-robot exploration and mapping. In Proceedings of the National Conference on Artificial Intelligence. AAAI

Thrun S, Bücken A (1996) Integrating Grid-Based and Topological Maps for Mobile Robot Navigation. In Proceedings of the Thirteenth National Conference on Artificial Intelligence

Yamauchi B (1996) A frontier-based approach for autonomous exploration. In Proceedings of the Thirteenth National Conference on Artificial Intelligence

Yamauchi B (1998) Frontier-based exploration using multiple robots. In Second International Conference on Autonomous Agents, pp 47-53

Zlot R, Stentz A, Dias M B, Thayer S (2002) Multi-Robot Exploration Controlled By A Market Economy. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)

Player/Stage website http://playerstage.sourceforge.net/