Improving Cooperative Robot Exploration Using an Hexagonal World Representation

Humberto J. Quijano and Leonardo Garrido
Centro de Sistemas Inteligentes
Tecnológico de Monterrey, Campus Monterrey
Av. Eugenio Garza Sada #2501 Sur, Monterrey N.L. Mexico, CP 64849

E-mail: {A00780887,leonardo.garrido}@itesm.mx

Abstract

In this paper, we used an alternative to the frequently used quadrangular grid map as representation of the world. This alternative uses an hexagonal grid instead a quadrangular grid to represent the world where simulated robots explore. Both representations used an occupancy grid map system, where the world is divided by many cells (hexagons or squares) and only one element can be in the cell at the same time. We made several experiments to observe how exploration efficiency is affected using both representations and four different exploration algorithms. Experiments were made in two scenarios. The first scenario tested the exploration efficiency of a single robot using different number of random obstacles. The second scenario tested the exploration efficiency using different number of cooperative robots leaving the number of random obstacles as a constant. We used SimER1 software[1] to make all exploration experiments in a simulation environment. Experiments results showed that hexagonal grid map representation was better than quadrangular grid map representation if the world is very big or if the exploration algorithm is not very efficient. Improvement in efficiency could be small if the exploration algorithm is very efficient, cooperation among exploration robots is very good, the area to explore is small or if the number of exploration robots is big. Nevertheless, even if improvement in efficiency is small, the variance in hexagonal representation is lesser. Reduction of variance in consecutive exploration experiments is an advantage because we are interested in explorations that always have a good degree of efficiency.

1 Introduction

In recent years, the use of cooperative robots to explore an unknown and dynamic area is an appealing idea. Its use has several advantages: faster exploration of an unknown area, complete theirs job even if some robots fails, cooperative robots can be cheaper if are used many simple robots instead a few or one expensive and complex robot.

The idea of a cooperative robot is a robot that can coordinate with others robots as a team. If one robot takes certain amount of time to accomplish a job, then two cooperatives robots should be able to finish the same task in the half amount of time or at least they should do a better job.

Coordination and team work of cooperative robots make possible that cheap and simple robots could do the same work of one expensive and very complex robot. Having more than one robot doing the same task is an advantage even if the cost of the set of simple robots and the cost of the complex robot is the same. If one of the group of simple robots fails, then we have others robots to finish the job and also the cost to replace the robot will be relatively cheap. Nevertheless, if the expensive and complex robot fails, then no one will finish the job and the cost to replace the robot could be very high.

2 Related Work

There has been much research about mapping and robot exploration using a single robot[2, 3, 4, 5, 6]. Nevertheless, there are not many papers about mapping and exploration using cooperative robots.

There are many techniques of how represent the world where robots will move. The most common representation is the quadrangular occupancy grid. It is simple and in most cases is enough to solve navigation problems.

0-7695-2974-7/07 $25.00 © 2007 IEEE
DOI 10.1109/CERMA.2007.72

Dungeons and Dragons[7] originally it was only a tabletop game that now uses both quadrangular and hexagonal grid maps to represent the area where the players are interacting. It began using a quadrangular grid. Nevertheless, due to diagonal move complexity, the game was updated to let players use an hexagonal representation instead a quadrangular representation. Hexagonal grid simplified the movement process. It is similar to our work because we began using quadrangular representation and then we changed to hexagonal representation. Nevertheless, all the work is based just in simplify movement process and not in movement efficiency as our paper.

In theirs paper, Yngvi Bjornsson, Markus Enzenberger, Robert Holte, Jonathan Schaeffer and Peter Yap[8], analyzed path finding for different kinds of grid maps representations, included hexagonal grid map. Besides, Gamedev.net[9] released an article, which compares characteristics of isometric and hexagonal grid maps used in video games to represent the world. Both works focus their research looking for the best and shortest path to the goal, but are not focus on other applications as map representations or exploration efficiency as our paper.

In his paper, Benedek Nagy[10] worked with hexagonal cubic grid representation instead a square cubic grid, where was developed new experimental formulas to calculate distances between points and neighbors in a hexagonal plane. This theory of neighborhood sequences is applicable in many image-processing algorithms.

There are also world representations which are more accurate representing the world than quadrangular or hexagonal occupancy grids. Nevertheless, the complexity of these representations increases a lot. Cyrill Stachniss and Wolfram Burgard[11]. In a quadrangular grid representation, instead representing each square as a free space or obstacle, each square has the right proportion of obstacle and free space. Robert Grabowski, Pradeep Khosla and Howie Choset[12] made in their paper a similar representation of the world using a new occupancy map system. Both representations has the advantage that could be represented almost all kind of irregular objects. Nevertheless, that kind of representations are not very simple to implement and also makes path finding more complex than just check if the area is free or not.

3 Hexagonal World Representation

Before simulated robots can be able to explore an unknown area, a representation of the world must be designed. One of the most common and simplest representations is the quadrangular grid map, where each square represents a small portion of the whole world.

In a quadrangular grid, diagonal distance between two squares is bigger than horizontal or vertical distance between two squares. Thus, it is harder to manage distances when robots move in diagonal, vertical and horizontal ways. To simplify that problem, robots frequently are restricted to only make vertical or horizontal moves when that kind of map representation is used.

In a hexagonal grid, diagonal and vertical distances between two hexagons are the same. Thus, there is no need to worry about managing different distances scales or restrict robots to only make vertical or horizontal moves. Figure 1 shows both grid map representations, hexagonal and quadrangular grid map. Both grids are representing the same map. Nevertheless, in the hexagonal grid, lines and curves can be better represented than in the quadrangular grid even if resolution is improved in both representations, where the map resolution is number of squares or hexagons used to represent the same area. Furthermore, distance between any two consecutive hexagons is the same and distance between two consecutive squares is not always the same.

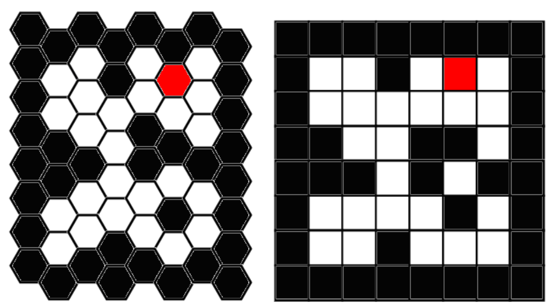

Figure 1. World representation using an hexagonal and quadrangular grid.

If we make an algorithm which calculates distance from a cell (hexagon or square) of the grid in both representations, we will get something similar to figure 2. The algorithm that makes this should work as follows: First, we represent the same world using both grid map representations. Then we select a seed. A seed is a coordinate “x” and “y” of the initial location, it must be a free space. This coordinate must be the same for both representations, for example coordinate 5, 5. Then we calculate distance from the seed to its neighbors. If the neighbor is an obstacle, the cell is ignored. Then we calculate distance from the seed to the neighbors of the seed’s neighbors. This process is repeated until we know distances to all cells of the grid.

If we observe figure 2, we can see that the cell with the number zero is our seed, our initial location. From the seed, each consecutive hexagon or square has the value of its neighbor incremented by one. At this point we can notice, for this example, distance from the seed to the most

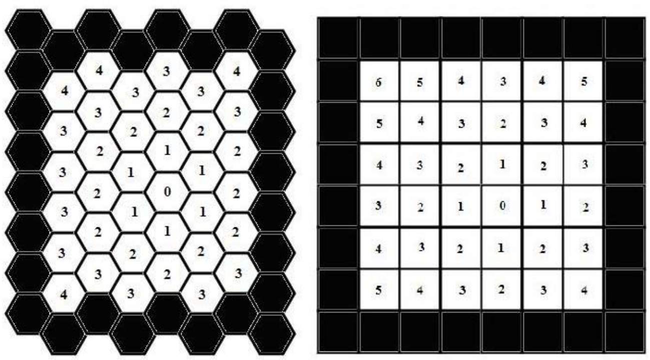

Figure 2. Distance factor in hexagonal and quadrangular representation.

remote location is four in hexagonal representation and six for quadrangular representation.

As we can see, hexagonal representations are better if we are interested in minimizing distance. Nevertheless, in practice, we need to consider other factors if we are interested in robot exploration. First, the exploration algorithm could make this advantage useless. Second, if the area is small enough or if it looks like a maze, distance is almost the same in both representations. Third, if the coordination among cooperative robots is very efficient or there are many exploration robots, then the improvement could be so insignificant that is unnoticeable or there is no worth in use hexagonal representation. Considering these factors, it is worth of make some experiments to get empirical results.

4 Simulated Robots

The goal of our simulated robots is to explore the whole area, making the least possible number of movements. The robots have a very limited perception of the world. They can only sense the cell (hexagon or square) which they are standing and the cell in front their line of vision. Nevertheless, our robots have an advantage, they have a radio communication that let them report their discoveries and location with other cooperative robots.

Our robots initiate in the simulated world knowing only its location and orientation in the world. Nevertheless, they are constantly trying to communicate and coordinate with other cooperative robots in the area via radio. If they found another cooperative robot, they coordinate to share its explored map and coordinate to explore certain areas. If one cooperative robot is going to explore an unknown area, the other robots do not try to explore the same area.

The used exploration algorithm is chosen by the user before the simulation is initiated. The exploration algorithms are described in detail in the next section.

In addition to the robot cooperation mechanism described previously, robots coordinate to trying to explore the whole map. If the way is blocked by other robot or an obstacle, then the cooperative robot coordinates with other robots to ask for help to explore the area. If it and the rest of simulated robots can not explore any more areas because are blocked or the whole area is explored, the exploration is finished.

5 Exploration Algorithms

An efficient exploration of world requires an efficient algorithm which makes robots explore the entire map making the less number of movements. Movements is the most expensive task for robots, it involves more time and energy consumption. In Artificial Intelligence: a Modern Approach[13] are described several search algorithms, which are similar to our version of exploration algorithms. In a previous work[14], we experimented with exploration efficiency of these algorithms using single and cooperative robots.

The exploration algorithms are versions of well known graph search algorithms. Each square or hexagon is represented with a node on that graph. We developed four different exploration algorithms which are:

Breadth-First Search[15]: This algorithm is a graph search algorithm that begins at the root node and explores all the neighboring nodes. Then, for each of those nearest nodes, it explores their unexplored neighbor nodes, and so on, until it finds the goal.

Breadth-First Search Random: This algorithm is a modified version of the breadth-first search algorithm. The algorithm does all the same like the breadth-first search algorithm except that it does not take the neighboring nodes to explore and expand. Instead of that, this algorithm selects a random element of the linked list to explore and expand.

Depth-First Search[16]: It is an algorithm for traversing or searching a tree or graph. Intuitively, one starts at the root (selecting some node as the root in the graph case) and explores as far as possible along each branch before backtracking. This process is continued until is explored all the area.

Best-First Search[17]: It is a search algorithm which optimizes depth-first search by expanding the most promising node chosen according to some rule. Examples of best-first search algorithms include Dijkstra's algorithm[18] and the A* search algorithm[19]. Best-first algorithms are often used for path finding in combinatorial search. Best-First search algorithm makes

ER1 robot first explores all the areas around it. After accomplished that, it moves a little bit and explores again all the unexplored areas around it.

6 Experimental Setup

In the Hexagonal World Representation section, we showed that hexagonal representation could be better than quadrangular if we are worried about distances. Nevertheless, we setup a couple of experimental scenarios in order to get empirical results with both map representations using different exploration algorithms. Besides, these results will give us additional information, like the variance in the experiments.

Experiments were made in two scenarios testing the efficiency of both map representations (hexagonal and quadrangular) using a 10X10 grid map area. To measure the efficiency of the exploration algorithms and map representation, we focused on the number of movements made by simulated robots. Movements is the most expensive task for robots, it involves the most energy and time consumption. If we minimize the number of movements made, then we make efficient the exploration.

In the first scenario, it was tested the exploration efficiency using a single robot with different number of random obstacles. It was divided in 6 categories which are: a field with no obstacles, with 10 random obstacles, 20, 30, 40, 50 and 60 random obstacles.

In the second scenario, it was tested the efficiency using different number of cooperative robots, leaving the number of obstacles constant in every experiment. Maps were generated with 30 random obstacles. This experimental scenario was divided in five categories which are: an exploration with one robot, with two robots, three, four and five robots.

To simplify the process, each square or hexagon represents the minimum space where a robot can stand. We defined that robots can only move in multiples of a constant distance, distance that is represented by a single cell (square or hexagon). Consequently, robots can move two or three squares but can not move 2.5 squares. In quadrangular grid representation, robots can only move north, east, south and west. In hexagonal grid representation, robots can only move north, north-east, south-east, south, south-west and north-west.

We made all exploration experiments with simulated ER1 robots using our SimER1 simulation software[1]. In each test, every simulated robot begins in a random location of the map, which must be a free space. Once a robot is initialized, only begins knowing its initial location, orientation and dimensions of the area that needs to explore. In order to give the scenarios a more realistic touch, we limited our simulated robots to only be able to sense the square or hexagon that is in front of it.

The world which robots will explore is a semi-static one, the only dynamic factor is the location and orientation of other exploration robots. The world could have three different elements which are: obstacles where robots can not pass, free spaces where robots can navigate and other exploration robots.

7 Experimental Results

In each exploration test, we registered three efficiency factors: total number of movements, total number of turns and total number of senses made by each robot. In these scenarios, we focused in the total number of movements made by robots to measure its efficiency. In the following subsections, we describe the results of both experimental scenarios: the single exploration one and the cooperative exploration one.

7.1 Single Exploration Scenario

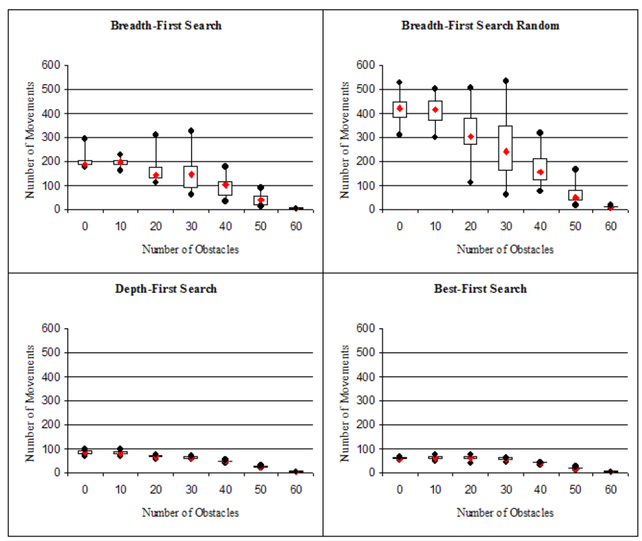

Figure 3. Box plot of single exploration scenario using quadrangular grid.

Figure 3 shows the box plot of number of movements by each exploration algorithm using quadrangular representation. Besides, figure 4 shows the box plot of number of movements using hexagonal representation. If we compare both figures, we can notice that number of movements made was lesser in hexagonal representation. This result confirms

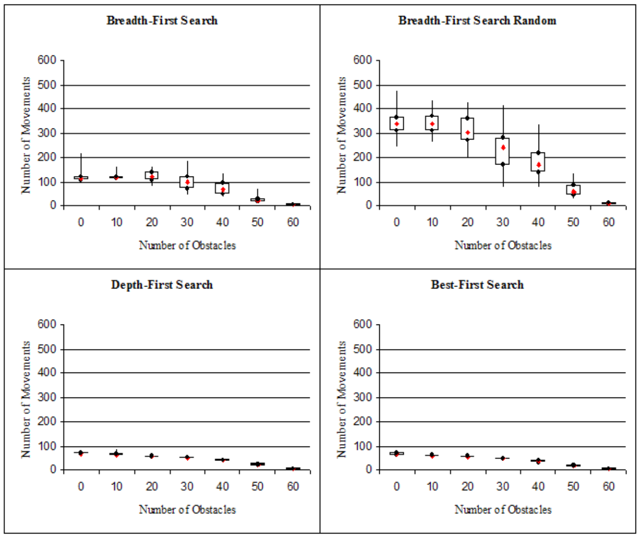

Figure 4. Box plot of single exploration scenario using hexagonal grid.

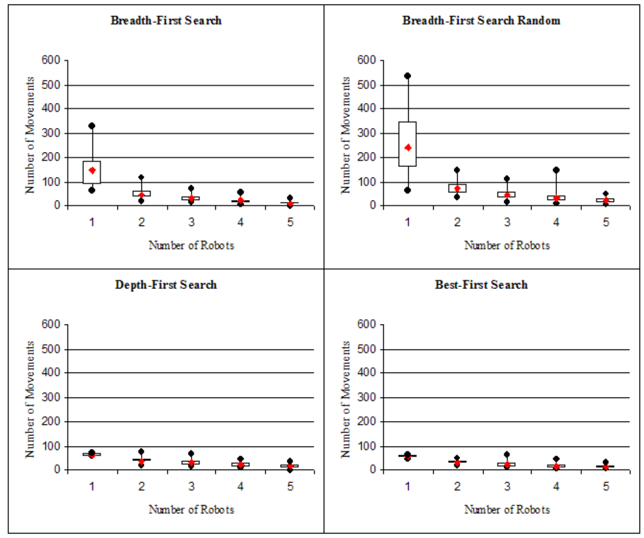

Figure 5. Box plot of cooperative exploration scenario using quadrangular grid.

our expectations. However, we also discovered that the variance was reduced significantly in hexagonal representation.

In the case of 20 to 40 obstacles, the random maps generated were very similar to a maze. Consequently, the exploration efficiency depended on where was the initial location of simulated robots.

7.2 Cooperative Exploration Scenario

Figure 5 shows the box plot of number of movements by each exploration algorithm using quadrangular representation. Besides, figure 6 shows the box plot of number of movements using hexagonal representation. If we compare both figures, we can notice that number of movements did not changed much in most of the cases. Cooperation among simulated robots made explorations very efficient. Besides, the efficiency increased with every simulated robot added into the area. Furthermore, hexagonal representation made that the variance was lesser than in quadrangular representation.

Robots in the experiments never hindered each other. Nevertheless, improves in the number of movement was lesser with each robot added. In the case of five robots, the cooperation made exploration so efficient that hexagonal representation did not improve the efficiency.

8 Conclusions

In this paper, we presented experimental results on cooperative robot exploration efficiency using an hexagonal and quadrangular grid map representation. Both representations used an occupancy grid map system, where the world is divided by many cells (hexagons or squares) and only one element can be in the cell at the same time. We made experiments, in two scenarios, with both map representation using several exploration algorithms. The first scenario exploration tested efficiency using a single and different number of obstacles. The second exploration scenario tested efficiency using different number of cooperative robots with a constant number of obstacles.

The results of our experiments showed that hexagonal grid map representation was better than quadrangular grid map representation if the world is very big or if the exploration algorithm is not very efficient. Improvement in efficiency could be very small if the exploration is very efficient because of an exploration algorithm, the map is very small, the number of exploration robots is big or because cooperation among robots is very good. Nevertheless, even if improvement in efficiency is very small, using an hexagonal representation makes the variance be lesser.

Reduction of variance in consecutive exploration experiments is an advantage. We are making explorations that always have a good degree of efficiency, instead of explorations that some times have a moderate degree of efficiency and good degree of efficiency in the rest of the times. For future work, we plan to make experiments using uncertain worlds where other agents modify or hinder our exploration

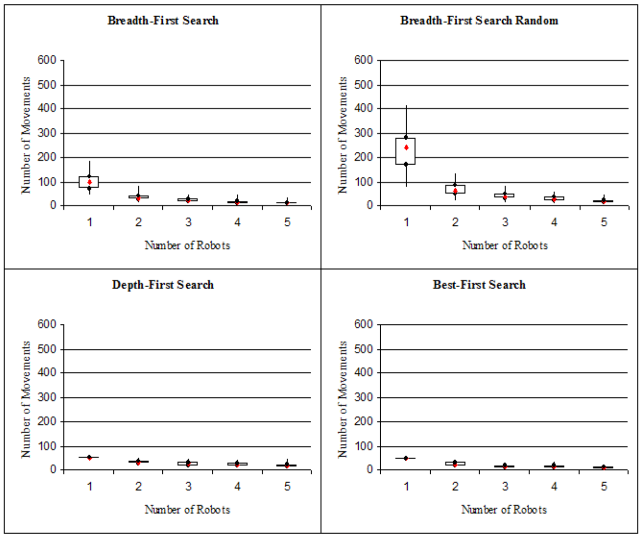

Figure 6. Box plot of cooperative exploration scenario using hexagonal grid.

robots. With this we want to compare how variance is affected using an hexagonal representation and quadrangular representation.

9 Acknowledgements

This work was supported in part by the ITESM research CAT011 on distributed knowledge and intelligent agents technologies.

References

[1] DELETED FOR BLIND REVIEW. (Technical report)

[2] Choset, H.: Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph. PhD thesis, California Institute of Technology (1996)

[3] Dudek, G., Jenkin, M., Milios, E., Wilkes, D.: Robotic exploration as graph construction. IEEE Transactions on robotics and Automation (1991)

[4] Kuipers, B., Byun, Y.: A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. Journal of Robotics and Autonomous Systems (1991)

[5] s. Thrun: Exploration and model building in mobile robot domains. In Proc. IEEE Intl. Conf. on Neural Networks (1993)

[6] Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence (1998)

[7] Gygax, E.G., Ameson, D.: Dungeons and Dragons. Wizard of the Coast (1974)

[8] Bjornsson, Y., Enzenberger, M., Holte, R., Schaeffer, J., Yap, P.: Comparison of different grid abstractions for pathfinding on maps. International Joint Conferences on Artificial Intelligence (IJCAI) (2003)

[9] TANSTAAFL: Isometric 'n' hexagonal maps. GameDev.net (1999)

[10] Nagy, B.: Calculating distance with neighborhood sequences in the hexagonal grid. Springer (2004)

[11] Stachniss, C., Burgard, W.: Mapping and exploration with mobile robots using coverage maps. Foundation for Research and Technology Hellas (FORTH). (2003)

[12] Grabowski, R., Khosla, P., Choset, H.: An enhanced occupancy map for exploration via pose separation. International Conference on Intelligent Robots and Systems (2003)

[13] Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, Inc (2005)

[14] (DELETED FOR BLIND REVIEW)

[15] Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. McGraw-Hill (2001)

[16] Black, P.E.: Dictionary of Algorithms and Data Structures. U.S. National Institute of Standards and Technology (2005)

[17] Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley (1984)

[18] Hart, P.E., Nilsson, N.J., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics SSC4 (1968)

[19] Nilsson, N.J.: Principles of Artificial Intelligence. Palo Alto, California. Tioga Publishing Company (1980)