Analysing Path Planning Algorithms for Mobile Robots

Joshua Daly Supervised by Arnold Hensman

Institute of Technology Blanchardstown
Institiúid Teicneolaíochta Baile Bhlainséir

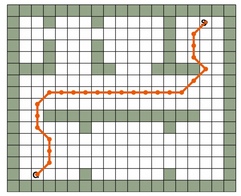
Background

Navigation is all about getting from A to B and for mobile robots this is challenging because

- environments are often complex and filled with obstacles
- there can be many paths to a desired goal but only one is optimal

Robots can tackle these difficult problems using grid based path planners which are

- simple and easy to implement
- widely used in robotic navigation



Above: an example of path produced by a planner through a grid environment

However a number of problems exist with traditional path planners like D* Lite

- they limit the number of headings a robot can take to increments of 45º
- in practice these paths can be longer, unnatural, and costly to traverse

Objective

To analyse the effectiveness of current grid path planners for mobile robots by comparing

- time taken to produce a path
- number of operations required
- length of the path and its traversal time

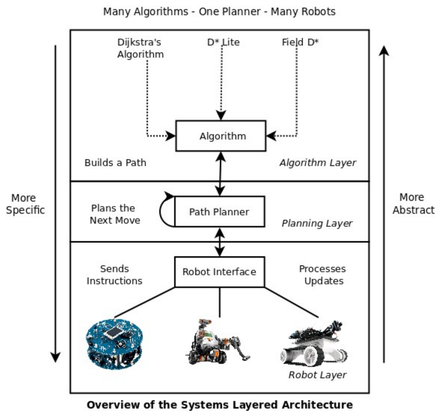
By looking at these attributes for D* Lite, GridNav, Field D*, and Theta* we can establish

- what is more important faster planning or shorter paths that take longer to compute
- best technique for heading calculation
- which algorithm consistently produces the shortest path

Implementation

The Rapid Application Development cycle enabled this project to progress quickly

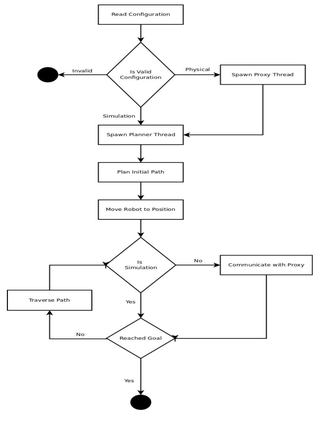
- application core implemented in *Python* with generic interface to robot platforms
- path planners *D* Lite, GridNav, Field D**, and Theta* integrated into system
- capable of gathering data results and processing them with the *gnuplot* program



Overview of the Systems Layered Architecture

Wealth of data gathered was processed to produce results that describe

- cost of traversing a planners path compared to the alternative paths
- practical effectiveness for real time applications on mobile robots

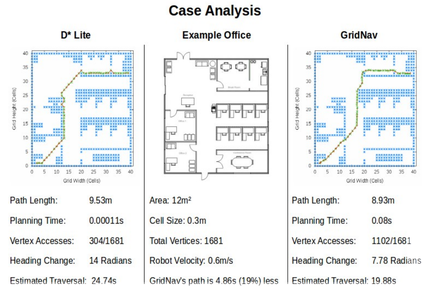


Above: process flow of the planning system from start to finish

Testing

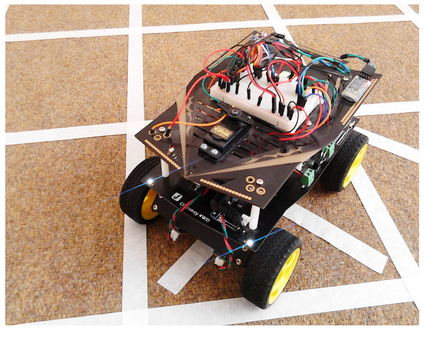
True analysis involves comprehensive testing and result gathering which requires flexibility

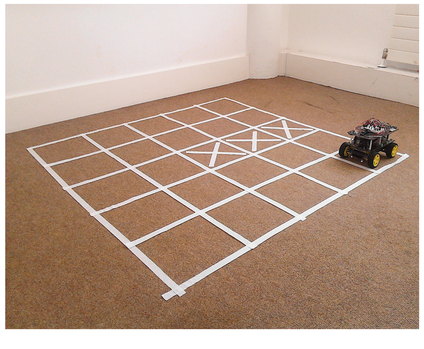
- uses abstract programming techniques to facilitate cross planner testing
- gathers data from both software simulated and physical hardware robots



Above: a comparison of the performance of both D Lite and GridNav in an example environment*

During the physical tests this 4WD DFRobot Cherokey powered by an Arduino Mega aided in confirming each planners effectiveness



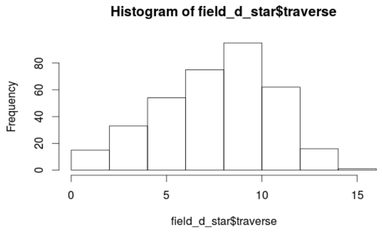


Results

During the testing cycle over 4,600 paths were traversed and evaluated

- each planner performed 1,200 traversals
- most results gathered from simulations, less prone to errors caused by hardware issues

Analysis was carried out using the statistical tool R which was used for graph generation



*Above: a histogram showing the traversal times for every path in an environment for Field D**

After analysing the results we discovered a number of interesting points

- vast majority of cases favoured planners that can deal with any heading
- computation time represented less than 1% of a traversal operation
- length of a path and rotations required has much greater impact on performance

Conclusion

Study found that planners which are not limited to a small set of headings can be up to 10% more efficient

Algorithms ranked from most efficient:

1. Theta*
2. GridNav
3. Field D*
4. D* Lite

Most promising planner was GridNav which produces smooth paths while remaining simple to implement

Joshua Daly, Institute of Technology Blanchardstown 2012-2015 | B00056835@student.itb.ie