Path Planning for a Mobile Robot in a Rough Terrain Environment

Dorian J. Spero

Intelligent Robotics Research Centre
Monash University,
Clayton, Victoria 3168, Australia
Email: dorian@ieee.org

Ray A. Jarvis

Intelligent Robotics Research Centre
Monash University,
Clayton, Victoria 3168, Australia
Email: ray.jarvis@eng.monash.edu.au

Abstract

The robust control of a mobile robot in a rough terrain environment is a challenging endeavour, since any reliance on favourable surface or environmental conditions will inevitably lead to task failure. This paper presents the preliminary development of a unified navigation system used to control a nonholonomic mobile robot in an a priori unknown outdoor domain. Accurate, high-resolution environmental data was gathered from a scanning laser rangefinder (ladar), which constitutes the robot's exteroceptive perception system. Using this data, a 3D tessellated environmental model was created that generically captures terrain traversability. By adapting the Rapidly-exploring Random Tree (RRT) approach to the tessellated model, an efficient kinodynamic path planning algorithm was devised that enables point-to-point trajectory traversal. This path planning strategy was found to be a computationally efficient method of producing robust and versatile path plans.

1 Introduction

Mobile robot navigation requires the successful integration of sensing, environmental mapping, localisation and path planning. These four critical components are governed by a control system to render the desired robotic function. In a rough terrain environment the control system needs to account for an undulating surface that has a cover taxonomy possibly consisting of rocks, trees, bushes, grass, soil, and water. The surface and environmental uncertainty existing in an outdoor domain makes the task of robot navigation a complex issue.

Rough terrain navigation has applications in cross-country and interplanetary exploration, search and rescue, and security missions (for example, ground surveillance). Mars exploration is an application domain where there is always a human presence in the control loop, however, due to the limited communication bandwidth and long delays, continuous teleoperation is not possible bringing about the need for a certain level of robot autonomy. Unlike indoor workplaces, the quest for distance optimality alone (as in A* search [1]) is insufficient in rough terrain as the complex and harsh nature of the environment cannot be disregarded. For instance, bodies of water or muddy patches might not be geometrically revealed to the perception system but will inevitably damage the vehicle if not taken into consideration. Consequently, path planning in an outdoor environment is at best a suboptimal arrangement of distance, time, energy consumption and safety factors.

Several techniques have been proposed for navigating in unstructured environments. In [2] a video camera is mounted at a high vantage point to visually servo the mobile robot. While pragmatically shown to be reliable, this method restricts the robot's configuration space to the camera's field of view. Artificial potential fields have been proposed [3], which model the goal point as an attractant and the obstacles as repellents. A continuous potential field is then computed to reactively guide the robot. However, this physical analogy commonly suffers from *cul de sacs* (concave obstacle configurations) that trap the robot.

The distance transform methodology [4] is a simple and robust technique used for both finding optimal collision free paths and obstacle growing (to accommodate the physical extent of the robot). The distance transform is calculated over a grid structured spatial representation. In the case of path planning, distances are propagated throughout each grid cell in an outwards direction from specified goal points, to ultimately fill the entire free space. Optimal paths are then found by using a steepest descent trajectory from any point in free space, without risk of local entrapment. Even though the distance transform has a generality (conferred by the tessellated model) that is useful in rough terrain environments, it is computationally expensive and does not scale well to large navigational areas at high resolution.

A versatile path planning strategy was proposed by LaValle and Kuffner [5], termed the Rapidly-exploring Random Tree (RRT). An RRT is a randomised data structure that is used to compute collision-free kinodynamic trajectories for high degree-of-freedom problems. To account for a robot's dynamic constraints a state space representation is used, which includes both configuration and velocity parameters. A path plan is generated by defining two RRTs, one rooted at the start state and the other at the goal state. Both trees are grown by first selecting a random state from the state space and then searching each tree for the nearest neighbouring state. All possible control inputs are applied to the neighbouring states to generate successor states that are collision-free, satisfy velocity bounds and minimise some chosen metric (for example, minimum distance) to the random state. The procedure is repeated until two states, one from each tree, are regarded as being sufficiently close in the state space to render a solution. The algorithmic pseudo code is listed in [5].

RRTs rapidly explore the state space and therefore scales well to large navigational areas, however, it is assumed that the environment is known *a priori* and is structured to assist with collision detection in real space. Also the nearest-neighbour search is a computationally expensive step. Hence, there is a need for a more efficient path planning algorithm that can robustly plan paths in an unknown rough terrain environment.

In this paper a path planning algorithm based on the RRT approach adapted to a grid-based environmental model is presented. The algorithm exhibits the beneficial properties of an RRT in an outdoor domain and naturally incorporates a computationally efficient nearest-neighbour strategy. A 3D tessellated mapping algorithm is proposed to effectively model a rough terrain environment. In order to build an accurate map, real-time environmental data was gathered from a scanning laser rangefinder (ladar). No *a priori* information is assumed.

In section 2, the perception system is described. Section 3 presents the grid-based environmental model used by the path planner proposed in section 4. Results from experimental tests of the system are reported in section 5. The paper concludes with a discussion of the proposed methods and future work in section 6.

2 Sensor System

The sensor selected to perceive the environment and produce a navigable map, was a laser rangefinder (ILM300HR). The laser was used to extract the scene geometry and since it is an active ranging system it can operate irrespective of ambient lighting conditions. The 905nm wavelength laser has a maximum range of 300m, a typical accuracy of 30cm and makes a measurement by using time-of-flight of a single laser pulse (single shot mode) at a repetition rate of 1000Hz. The laser is mounted on a high-speed pan-tilt unit (PTU-46-17.5) for accurate 3D positioning, as depicted in Fig. 1.

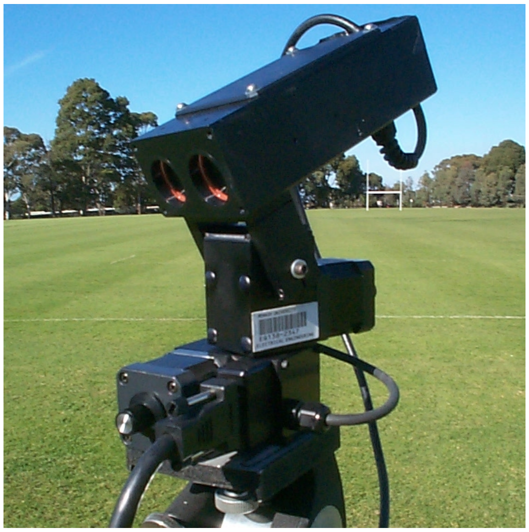


Figure 1: Laser Sensor and Pan-Tilt Unit

Range readings are taken in polar coordinates as $r = f(\theta, \phi)$, based on the reference system given in Fig. 2. The pan-tilt unit provides a 318° horizontal vision field ($\theta \in [-159°, 159°]$) and a 77.7° vertical vision field ($\phi \in [-46.7°, 31°]$).

The horizontal and vertical angular resolution is 0.0514°, determined by the gear ratios and angular granularity of the internal stepper motors. In the present investigation, for timely 3D data acquisition, the pan-tilt unit was set to a particular elevation angle and then horizontally scanned, while concurrently matching the timing of the laser readings to the adjustable scan speed profile. Consequently, the horizontal angular resolution had a variable granularity.

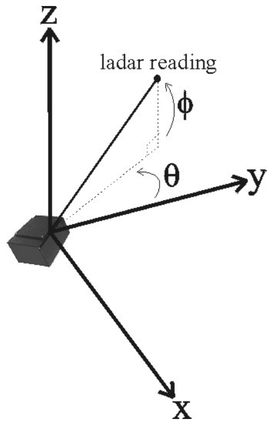


Figure 2: Ladar Reference System

Possible range outliers were eliminated using a moving median filter applied during a horizontal scan. A filtered range (r_f) was found by averaging three consecutive range samples, as shown in Eq. 1. For map building, the filtered polar coordinates were converted to cartesian coordinates, $z = f(x, y)$, using the transformation in Eq. 2.

$$r_f = \mathrm{m}edian(r_{n-1}, r_n, r_{n+1}) \tag{1}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -r_f \sin(\theta)\cos(\phi) \\ r_f \cos(\theta)\cos(\phi) \\ r_f \sin(\phi) \end{bmatrix} \tag{2}$$

3 Map Building

A tessellated map was developed that consists of a 1000x1000 cell lattice with a cellular resolution of one square meter, providing a total of one square kilometer coverage. Each cell contains a dynamic layer used to represent height (z) discontinuity in the terrain. The modeling of height discontinuity allows the robot to traverse underneath obstacles when geometrically able. For instance, the robot can pass beneath a tree's branches or a manmade structure if they are sufficiently high above the ground to pose no risk. In an indoor environment the z coordinate is commonly truncated at a particular height to achieve the same outcome. However, in an undulating outdoor environment, posing an artificial height limit could trap the robot in a surface depression (or valley) caused by the discarding of important surface data.

To adequately map a rough terrain environment, each cell contains three fields (Fig. 3): the maximum obstacle height (z_{max}), the minimum obstacle height (z_{min}) and the terrain composition. The z_{max} and z_{min} parameters track the effective occupied volume within a cell. The terrain composition field is for load-bearing surface determination and as it is a topic of ongoing research, as discussed in section 6, it is only mentioned here for completeness.

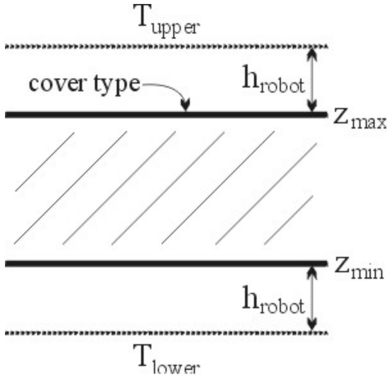


Figure 3: Internal Cell Structure

The z_{max} and z_{min} values form an obstacle layer that dynamically adjusts to incoming sensor data. If the robot can traverse a cell at z_{max} or under z_{min}, the cell is considered free space. Each 3D sensor value, $z_i = f(x_i, y_i)$ indexed by $i = 0..i_{max}$, is inserted into the map by first locating the relevant cell via the x_i and y_i values and then using the z_i value to determine the affect of the range sample on the internal structure of the cell. The procedure for updating a cell's obstacle layer is described by the pseudocode listed in Fig. 4.

```
UPDATE_CELL( $z_i$ )

//define upper and lower threshold
$T_{upper} = z_{max} + h_{robot}$;
$T_{lower} = z_{min} - h_{robot}$;

if cell unexplored or $z_i < T_{lower}$ then
    set $z_{max}$ and $z_{min}$ to $z_i$;
else if $z_i > z_{max}$ and $z_i < T_{upper}$ then
    $z_{max} = z_i$;
else if $z_i < z_{min}$ and $z_i > T_{lower}$ then
    $z_{min} = z_i$;
else
    discard data point;
```

Figure 4: Algorithm for Updating $Cell_{x,y}$ with a New Ladar Value

Using the robot's height (h_{robot}), the T_{upper} and

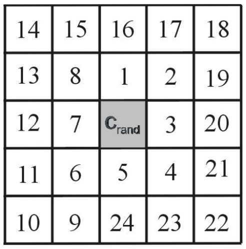
T_{lower} thresholds are calculated to obtain the minimum height above and below the obstacle layer respectively, for which the robot is geometrically able to pass through. If new data points are found to be within the threshold limits, the obstacle layer expands to consume each point to record the change in the robot's viable passageways. Data points above T_{upper} are discarded as they do not impinge on a cell's free space, however, data points below T_{lower} causes the obstacle layer to shift down to effectively target the obstacle's base, which is more pertinent to a ground vehicle. This process iteratively builds the grid-based environmental model for use by the path planner.

4 Path Planning

The proposed path planning strategy generates point-to-point kinodynamic trajectories using the tessellated map discussed in section 3. The path planning problem is treated as a search in state space, X, that incorporates both configuration and velocity parameters. A collision-free path that satisfies velocity bounds (as determined by the robot's locomotion system) lies entirely in X_{free}, where $X_{free} = X \backslash X_{obs}$. States in X_{obs} includes obstacle regions, velocity constraints and states of inevitable collision where the robot, due to its velocity, cannot perform collision avoidance. Unexplored regions are optimistically assumed to belong to X_{free}.

Two Rapidly-exploring Random Trees (RRTs) are defined, T_{start} and T_{goal}, initialised to the start state (x_{start}) and goal state (x_{goal}) respectively. Both states correspond to a cell in the map and a control input, u, indicating velocity. Each tree consists of a set of states, or vertices, linked by edges that represent feasible paths in the environment.

The RRTs are grown by first selecting a random cell (c_{rand}), constrained only by the grid boundary, followed by searching for the nearest neighbouring cell contained in T_{start} and T_{goal}, which initially would be x_{start} and x_{goal} respectively. The nearest-neighbour search is performed by picking a random cell (c_1) at a distance of one cell from c_{rand}. If cell c_1 is a vertex in the target tree, the search stops. Otherwise the remaining cells, at the same distance from c_{rand}, are checked in a clockwise direction from c_1. The process repeats at sequentially increasing distances from c_{rand}, until the nearest neighbouring vertex is found. As depicted by Fig. 5, the order of the search propagates in an outward direction from c_{rand}, analogically resembling a water ripple.



14	15	16	17	18
13	8	1	2	19
12	7	c_{rand}	3	20
11	6	5	4	21
10	9	24	23	22

Figure 5: Nearest-Neighbour Search

Control inputs are applied, for a fixed time interval (Δt), to the nearest neighbouring cells. The inputs are chosen such that the created edges are in X_{free} and the Euclidean distance to c_{rand} is minimised. Each newly created edge is finally terminated by a vertex. Repeating the procedure iteratively expands T_{start} and T_{goal}, adding feasible vertices and edges during each cycle.

The RRTs are grown throughout X_{free} until a cell in T_{start} intersects, or is directly adjacent to, a cell in T_{goal} providing a point-to-point path between x_{start} and x_{goal}. An upper limit of K vertices is imposed, in case x_{goal} is unreachable.

5 Experimental Results

Experimental tests were performed in an outdoor environment using a 3D laser scan, which covered a horizontal range of $[-154.2°, 154.2°]$ and a vertical range of $[-25.7°, 20.6°]$. The horizontal angular resolution was set to 0.0771°, with the vertical angular resolution set to 0.257°. This resulted in approximately 725000 data points being gathered in a 10 minute time period. The filtered data from a laser scan, shown in Fig. 6, proved to be sufficiently accurate to map the outdoor testing area.

The proposed path planning algorithm was initially tested using an unexplored grid, to observe its behaviour under a vertex limit (K) set to 1000, a control input of up to $20cm/s$ and a Δt of $30s$. As depicted in Fig. 7, the RRTs rapidly expand to the grid boundaries providing efficient feasible path generation. The use of two RRTs, one rooted at x_{start} and the other at x_{goal}, was found to give a higher probability of intersection between the start and goal cells.

An indication of the computational efficiency of the nearest-neighbour search is given in Fig. 8, which was obtained using a single RRT with K set to 10000. As shown, the number of cells searched per cycle decreases exponentially as the total number of ver-

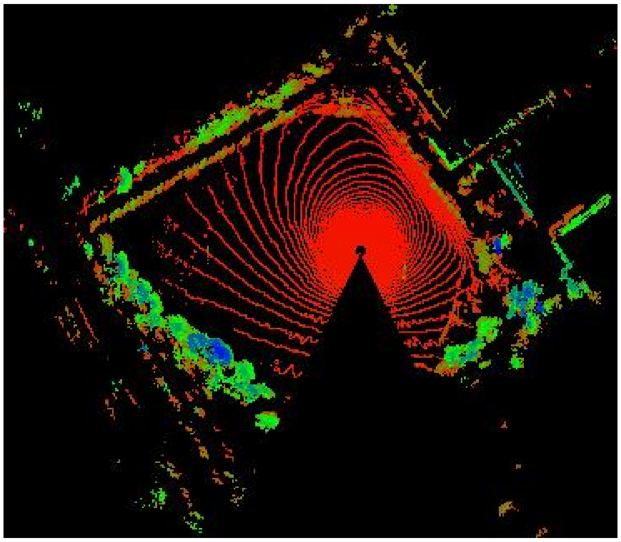


Figure 6: Laser Scan of an Outdoor Environment

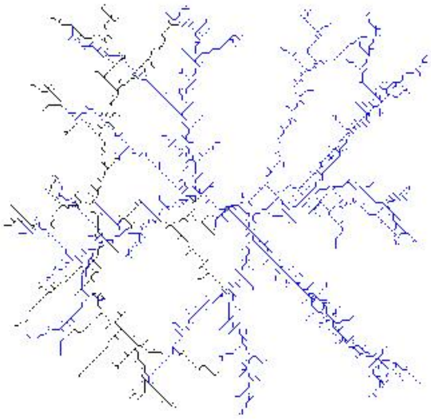


Figure 7: Rapidly-Exploring Random Trees Adapted to the Tessellated Model

tices generated increases. This beneficial property translates into a decreasing computational load as the RRTs dilate over time.

Fig. 9 depicts a path planning scenario where a path has been generated from x_{start} to x_{goal} using the map created from the laser scan (Fig. 6). The path traverses across a rugby field, circumnavigates clusters of trees and heads down a road to reach the target. Even though the path plan is not globally optimal, with respect to distance, it is efficiently generated and feasible for an outdoor environment.

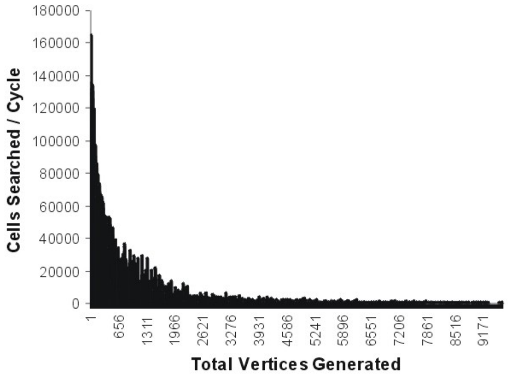


Figure 8: Computational Efficiency of the Nearest-Neighbour Search

6 Discussion and Conclusions

The presented path planning approach was found to efficiently produce feasible paths in an outdoor environment. The generated paths incorporate dynamic constraints, accommodating the robot's locomotion capabilities or lack thereof. The RRTs rapidly explore the free space by a process of random sampling, tending to extend the vertices within relatively large Voronoi regions. Due to the randomised nature of the algorithm, problems due to *cul de sacs* have not been found, however, more investigation is required for absolute confirmation.

Generated paths were found to incur small zigzags, or random noise, which could cause the robot to turn unnecessarily. Using a filter, such as turning only when the path deviates outside a specified channel, can easily eliminate this problem and therefore minimise the number of rotational movements, if translational movements are favoured.

The proposed path planner can efficiently handle updates to the environmental map, as the robot gathers new sensor data from different vantage points. A cell considered free but then later found to be occupied, can be counteracted by simply discarding any paths traversing through that cell. Conversely, a cell assumed to be occupied but then later found to be free, does not require any counter-action as the cell becomes a member of free space and is naturally incorporated into future path planning decisions. The start state does not need to be redefined, or the algorithm rerun, upon map changes.

Another beneficial property is that the path planner can provide alternative paths, giving the robot the

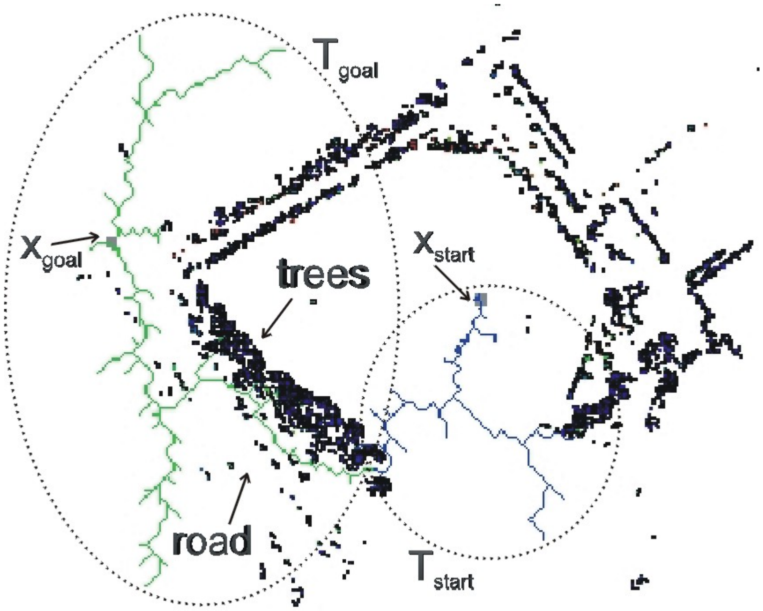


Figure 9: Path Planning Scenario

flexibility to choose the best path based on an optimal criteria. Instead of stopping the algorithm at the first solution, the RRTs can be continually grown until the desired solution is yielded and possible contingencies addressed.

An accurate vehicle model is currently being devised to test the path planner on a converted outdoor vehicle. For a complete rough terrain solution, future work will focus on the following areas:

- localisation based on natural landmarks, where the robot can robustly determine its position without the need for site preparation (for example, inserting beacons).
- load-bearing surface determination to differentiate between the surface touched by the robot's wheels and the visible surface, as discussed in [6, 7].

References

[1] N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill, New York, 1971.

[2] R. Jarvis, "Visually servoing an outdoor mobile robot," in *Conference Proceedings DICTA-93 Digital Image Computing: Techniques and Applications*, Sydney, Australia, 8-10 Dec. 1993, pp. 130–138.

[3] L. Montano and J. R. Asensio, "Real-time robot navigation in unstructured environments using a 3D laser rangefinder," in *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems*, Grenoble, France, 7-11 Sep. 1997, pp. 526–532.

[4] R. Jarvis, *Recent Trends in Mobile Robots*, vol. 2, chapter Distance Transform Based Path Planning for Robot Navigation, pp. 3–31, World Scientific, NJ, 1993.

[5] S. M. LaValle and J. J. Kuffner Jr., "Randomized kinodynamic planning," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI, May 1999, vol. 1, pp. 473–479.

[6] R. Manduchi, L. Matthies, and F. Pollara, "From cross-country autonomous navigation to intelligent deep space communications: Visual sensor processing at JPL," in *11th International Conference on Image Analysis and Processing*, Palermo, Italy, Sep. 2001, pp. 472–477.

[7] J. Macedo, R. Manduchi, and L. Matthies, "Ladar-based terrain cover classification," in *Unmanned Ground Vehicle Technology III*, Orlando, FL, 16-17 Apr. 2001, vol. 4364, pp. 274–280, SPIE.