Path Planning for Planetary Surface Exploration Using Incremental A-r-Star Pathfinder

Daniel Opoku
Autonomous Control and Information Technology Center
Electrical & Computer Engineering Department
North Carolina A & T State University
Greensboro, NC USA
do_opkuofficial@yahoo.com

Edward W. Tunstel
Johns Hopkins University
Applied Physics Laboratory Research & Exploratory Development Department
Laurel, MD 20723 USA
edward.tunstel@jhuapl.edu

Abdollah Homaifar
Autonomous Control and Information Technology Center
Electrical & Computer Engineering Department
North Carolina A & T State University
Greensboro, NC USA
Homaifar@ncat.edu

Abstract - This paper addresses rover path planning for wide-area exploration on remote planetary surfaces using the Incremental A-r-Star pathfinder algorithm. Given an initial terrain map derived from satellite imagery or images captured by a lander during descent as well as onboard 3D range sensors, a rover plans a safe and suboptimal path from location to location and through boulder fields while avoiding mobility hazards. Initial paths are based on the knowledge base obtained from the initial map but as the rover navigates and discovers new obstacles that were not initially represented in the map, it re-plans a new path from its current location. This results in reduced computation as compared to re-planning an entirely new path and hence increases rover navigation rate and reduces power consumption.

Keywords: path planning, rover navigation, planetary surface exploration, graph search algorithm, terrain map.

1. Introduction

The exploration of planetary surfaces has been facilitated by progressive advancements in planetary rover mobility and navigation technology. Algorithmic products of research in artificial intelligence (AI) for solving graph- and grid-based search problems have been applied as path planning solutions for rover navigation on the moon and Mars. As the required durations of surface missions increase along with rover capabilities and design lifetimes, future rovers will traverse larger areas of terrain, thus encountering larger variations in terrain type and topology.

Amongst the various terrain types identified on the moon and Mars are boulder fields with denser distributions than terrain at preferred landing sites or through which existing rovers have traversed on planetary surface missions to date. Unmanned spacecraft, robotic landers and rovers as well as astronauts of Apollo lunar missions have imaged boulder fields amongst other navigable but less complex terrain. They are known to exist on the surfaces of Venus, Earth and Mars with various size and spatial distributions as well as on Earth’s moon, Mars’ moon Phobos, and Saturn’s moon Titan, not to mention on asteroids [1]. Boulder field locations include terrain associated with domes, mounds, and lava fields as well as crater rims, walls, and floors. Considering that future high-value science targets or resources supporting human exploration may exist in or may be accessible only by traversing through boulder fields, it is important to establish suitable motion control and planning algorithms that would enable rovers to negotiate boulder fields.

To date, Mars rovers have employed global path planning algorithms such as Field D* [2] (and local path planners such as GESTALT [3]), which is a derivative of D*, a version of the classic A* path planning algorithm produced by AI research. China’s lunar rover, Yutu, uses a simple, robust particle swarm based algorithm for its global waypoint navigation planning [4]. Field D* has proven effective at negotiating both benign and complex terrain types encountered thus far on Mars by NASA rovers Spirit, Opportunity, and Curiosity. Other path planners for mobile robot navigation have been developed and applied to navigation tasks both indoors and outdoors for Earth-based mobile robots [5]. In recent work [6], an alternative path planner, Ar* (pronounced A-r-Star), has been shown to have certain advantages [7]. It is a modified version of the A* pathfinder in a uniform grid world that outperforms A* path planning in a sparse uniformly gridded world and matches A* in a cluttered world. A later version referred to as Incremental was enhanced for faster re-planning using information from past paths searched to speed up subsequent path searches [8]. The Incremental A-r-Star algorithm has broad applicability across indoor and outdoor areas and is presented herein as a viable option for path planning through boulder fields on planetary surfaces.

This paper is organized as follows. Section 2 provides an overview of the Incremental A-r-Star pathfinder. Section
3 describes the terrain map representation suitable for path planning through boulder fields. Section 4 presents an example application of the Incremental A-r-Star planner and its effectiveness supported by simulation results. Section 5 concludes the paper.

2. Incremental Path Planner

The Incremental A-r-Star algorithm presented in [8] is a dynamic version of the A-r-Star algorithm [6, 7]. This suboptimal, effective and efficient algorithm has been used for indoor path planning and earlier research has proven its powerful ability to deal with high resolution wide-area path planning without exponential increase in computational time [7]. The authors have compared the A-r-Star algorithm’s performance with that of other pathfinders in earlier publications [6, 7]. Also, the authors have enumerated with proofs [7], the properties and performance of the A-r-Star and the Incremental A-r-Star algorithms, including but not limited to: Completeness, Efficiency, etc.

The Incremental A-r-Star algorithm uses two major stages of planning, the offline stage called the process-state stage builds a direct acyclic graph (DAG) out of a tessellated regular grid of a finite region of the navigation environment. The DAG represents the navigable spaces and the relationship between nodes (grid cells) within. This relationship encodes the distance from one node to another in a parent-child relationship. The process-state stage performs the same task as the A-r-Star algorithm with the exception of originating from the target node (goal/destination) and planning towards the origin (start) node. The process-state stage takes a static obstacle map of the navigation environment and plans a path connecting the start and destination nodes. This map can represent an area imaged from a satellite, an area captured by lander camera images prior to landing of the rover, or a map from previous exploration. It may therefore only partially cover terrain areas of interest and may not include all surface obstacles/hazards due to camera resolution or occlusions. Besides, the initial map may have changed due to various environmental dynamics.

The second stage is the prune-branch stage. Given the DAG formed from the process-state stage, the prune-branch removes a parent node and all its descendants from the DAG. This begins from the parent and proceeds down the branch until the state of all the nodes are reset to unknown nodes. It however preserves the remainder of the DAG branches. When the rover discovers a change in its environment such as an occupied node initially perceived to be unoccupied, the prune-branch stage is executed to remove that node and its descendants from the graph. The process-state is then recalled to plan the path from the current stage to the closed node which is part of the DAG. Since each branch represents a suboptimal path to the destination, the next discovered path will be a suboptimal path. However, the re-planning time is reduced considerably since the previous path information is reused [8].

3. Terrain Map Representation

Figure 1 shows images of boulder fields on Earth’s moon and on Mars. The Incremental A-r-Star pathfinder can be applied to plan paths in such multi-level terrain with undulated topology including distributed features of both positive and negative elevation (e.g., representing boulders, rocks, mounds, holes, etc). However, to obtain a digital model useful for path planning, a reasonable assumption is to represent such multi-level terrain by a threshold model. That is, the multi-level terrain can be quantized into a binary, tessellated terrain representation wherein non-navigable terrain grid cells are labeled with zeros and navigable grid cells are labeled with ones. This is an option of common practice in the application of path planning algorithms for mobile robot navigation.

![Figure 1. Representative boulder fields: (top) boulder field dubbed “Kirkland Lake” on Mars (Credit: NASA/JPL-Caltech/Cornell/S. Atkinson); (bottom) boulders around Camelot Crater on the moon (Credit: NASA).](image)

Such abstraction of the topological complexity of the terrain must consider the mobility limitations of the rover intended to traverse the terrain. That is, a particular topological portion of terrain of area comparable to a given rover’s footprint is only considered traversable by that rover if it can be traversed without causing the rover to tip over (due to exceeding its maximum tilt limit for stability) or high-center (due to peaks of terrain that would raise the rover chassis such that its rover wheels/legs are suspended without contact above the terrain). Other factors such as a rover’s potential to become embedded or stuck in terrain (due to inability to maintain traction) could be applied to consideration of terrain traversability. Considering the mobility limitations for a given rover, tessellation elements of a particular terrain region can be treated as non-navigable or navigable for the purpose of the path planning terrain map representation. Then the task of planning a path
through free space within the map representation becomes one of path planning through its navigable space.

It is worth noting an effect of increasing grid cell resolution. A low resolution or coarse grid results in fewer cells to search, and hence permits faster path planner run time.

![Grid resolution effect on graph search algorithms](image)

Figure 2. Grid resolution effect on graph search algorithms

However, coarse tessellation of the continuous world can result in the creation of unrealistic paths where an obstacle is too close to a map boundary (see Fig. 2a-2c) and it can also place a path waypoint or goal location beyond reach, especially in a cluttered environment (see Fig. 2d-2f).

For recent Mars rovers, the finest terrain map grid cell resolutions used by surface navigation algorithms are comparable to rover wheel size. Larger grid cell sizes could be used to gain greater computational processing efficiencies or are justified when the terrain is fairly benign rather than cluttered with mobility hazards (that are comparable to wheel or rover footprint size). In this paper, the grid cell resolution is based on the rover as specified in the path planning example presented in the next section.

4. Example and Results

In a hypothetical mission scenario, local/regional maps are derived from imagery taken by satellites/orbiters and/or from cameras on landers as they were approaching the surface during descent. The maps are thus considered to cover areas of hundreds of square-meters to approximately one square-kilometer and to contain or span boulder fields. As part of a mission’s surface exploration activities the rover's goal is to navigate through such boulder fields using the Incremental A-r-Star algorithm. On a larger scale, the algorithm is also applicable to path planning through traversable terrain around distributions of non-navigable landforms (e.g., craters, hills, crevasses, etc) in the terrain map that are typically resolvable in images acquired by orbiter/lander-descent cameras. Paths for rover surface navigation beyond such a map would be planned within another map of similar size derived from images covering the next large patch of local/regional terrain of exploration interest.

Within the above scenario, we present an example application of the Incremental A-r-Star pathfinder and its effectiveness, as predicted by computer simulation, for planning paths in a representative boulder field on a planetary surface.

4.1. Rover and Terrain Map

For sake of demonstration in this work we consider the research rover prototype shown in Fig. 3, with a 61 cm long by 41 cm wide footprint, 11 cm wheel diameter, and 26 cm-high ground clearance. This prototype is further described in an earlier research paper [9]. A sample terrain map is shown in Fig. 4 wherein the boulder distribution is based on an analytical model derived from rock size and frequency distributions at the NASA Viking Lander sites on Mars [10]. A 3D view of the sample terrain is shown in Fig. 5. (In Figs. 4 and 5, only boulders large enough to be considered obstacles/mobility hazards are shown.)

![Research rover prototype](image)

Figure 3. Research rover prototype.

![2D view of terrain map](image)

Figure 4. 2D view of terrain map.
4.2. Path Planning Task and Results

This section presents a path planning simulation on the 2D terrain presented in Fig. 4. This local map is used for easily visualized demonstration. It is expected that successful path planning in such representative local boulder fields would also be successful in areas of larger regional extent with comparable obstacle distributions. The resolution of the terrain was tessellated into 700 by 700 cells. The dimension of each cell is 1/70 m. A clearance of 10 cells was allowed, representing 14.29 cm, which is greater than half the length of the shorter side of the rover, and thus greater than the rover footprint, to ensure path traversal within safe distances to obstacles.

The initial task involves planning a path from cell [650, 680] at 2D approximate position (9.29 m, 9.71 m) to goal cell [90, 40] at approximate position (1.29 m, 0.57 m) using the initial 2D terrain map shown in Fig. 4. The initial planned path, from the start location to the goal location, resulting from running Incremental A*-Star for this task is shown in red in Fig. 5. When the rover begins moving along the initially planned path, at position (8.07 m, 7.39 m), its onboard vision system (e.g., laser range finder or stereovision camera) detects an obstacle of size 0.29 m x 0.29 m which was not factored into the initial path planning, so the prune-state stage of the algorithm is executed to remove the affected cells and their descendants from the DAG and re-plans the new path using the remaining DAG (blue line on Fig. 7).

While navigating the new path, the rover observes another obstacle from position (5.37 m, 3.64 m) of size 0.43 m x 0.43 m which was also not known initially, the rover then re-plans the path around this obstacle as well and continues to navigate to its goal destination using the initial path. None of these re-planning exercises had to be done all over again since the DAG from a previous planning task is used for the next, and so the time and computational resources used are reduced. Time comparisons for A*-Star path planning with and without reuse of the previous DAG, and for A*-Star re-planning vs. a popular incremental search algorithm, D* Lite [11], are reported in [8].

Figure 5. 3D view of terrain map with rover (and boulder elevations depicted as 3D Gaussian peaks).

Figure 6. The initial Path Planning Task to plan from cell [650, 680] to the goal at cell [90, 40].

Figure 7. The second Path Planning Task to re-plan from cell [565, 517] to the goal at cell [90, 40].

Figure 8. The third Path Planning Task to re-plan from cell [376, 255] to the goal at cell [90, 40].
5. Conclusion

The Incremental A-r-Star algorithm can be applied to plan paths through boulder fields encountered during the exploration of planetary surfaces. Its ability to plan suboptimal paths based on partially known maps and to adjust the path incrementally with reduced time and computation makes it valuable for planetary surface exploration tasks, which are characterized by limited or otherwise constrained mission operations timelines and rover onboard resources (computation, power, allowable execution time, etc). This paper presented an application of the Incremental A-r-Star path planning algorithm to rover navigation task execution on a 2D grid map. Future work will explore the implementation using 3D maps.

Acknowledgment

I greatly appreciate the matchless support of my Ph.D. advisor Dr. Homaifar for his guidance and support during and after my studies. A big thank you goes to my Ph.D. co-advisor, Dr. Edward Tunstel, for his support during my studies and also his great contribution to the structure and content of this paper.

References


