COOPERATIVE SYSTEMS IN MISSION PLANNING

Pablo Muñoz and María D. R-Moreno

Departmento de Automática, Universidad de Alcalá, 28871 Alcalá de Henares, Spain
Email: {pmunoz, mdolores}@aut.uah.es

ABSTRACT

The complexity of future exploration missions encloses hard constraints such as driving the vehicles to many locations in a surface of several kilometers and collect/analyze interesting samples in challenging locations. Then, how to optimally (or close to optimal) plan the paths that the vehicle should follow having in mind the terrain features (slope, roughness, etc) and the energy consumption while performing the scientific tasks is critical for the mission.

In this paper we present a new planner that interleaves path planning with the traditional task planning capabilities. This planner is able to plan paths having in mind the terrain features and generate (close to) shortest paths with smoother direction changes while performing the scientific tasks in an efficient "ordered" way. To test our system, we have performed some experiments on two landing sites on Mars of two previous rovers exploration missions. The new system is able to handle maps grids ten times bigger than AI planners that do not integrate path-planning capabilities. Besides, the paths generated take into account the Digital Model of the Terrain to avoid high hills or hazard levels of rotation.

1. INTRODUCTION

Future ESA science missions such as ExoMars and MARS Sample Return, that are part of the Aurora program of the European Union Council of Research, will require of increasingly capable systems to achieve the goals they are designed for. In some cases these capabilities can be met by adapting or further developing existing technology, but in other cases new innovative technology should be injected in order to exploit science opportunities and reduce overall operation costs. Since the mobility and sciences capabilities of the rovers are increasing, more powerful tools to assist the operators on ground and autonomous capabilities for the rovers are required. For example, the Curiosity rover can travel up to 90 meters per hour on its six-wheeled rocker-bogie system. Therefore, how to optimally (or close to optimal) plan the paths that the rover should follow having in mind the terrain features (slope, roughness, etc) and efficiently manage the energy consumption while performing the scientific tasks is going to be critical for the future missions.

In this direction, the paper presents some of the results obtained within the Ph.D. program on the topic of *Cooperative systems for autonomous exploration missions* funded and supported by ESA. We have extended the deliberative layer of a 3T architecture [1] to interleave path planning with task planning capabilities. This new layer is able to plan paths having in mind the shortest path with smoother direction changes and terrain features while performing the scientific tasks in an efficient "ordered" way. We have called the integrated planner as UP2TA.

The paper is structured as follows. First section reviews the main classical path planning algorithms. Then, we describe the constraints that path-planning algorithms should meet when working on planetary surfaces. Next, the integrated planner is described. After, an experimental evaluation on Martian maps is presented. Finally, conclusions are outlined.

2. PATH-PLANNING REVIEW

The path-planning problem, that is, how to obtain a feasible route between two points, as a search tree over a discrete environment with blocked or unblocked squares has been widely discussed [2]. There are two types of representations: the nodes can be in the center of the grid (center node representation) or the nodes can be on the corner of the grid (corner node representation). For both cases, a valid path is that starting from the initial node reaches the goal node without crossing a blocked cell.

A node is represented as a lowercase letter, assuming p a random node and, s and g the start and goal nodes respectively. Each node is defined by its coordinate pair (x, y), being x_p and y_p for the p node. A solution has the form $\{p_1, p_2, ..., p_{n-1}, p_n\}$ with initial node $p_1 = s$ and goal $p_n = g$. For each node p we save: (1) the parent of the node, (2) the accumulate cost G(p), that is, the length of the shortest path from the start node to the p node and (3) the heuristic value or the estimation of the distance from p to the goal node, H(p).

Algorithms such as A* [3] allow us to quickly find routes at the expense of an artificial restriction of direction changes of $\pi/4$. However, there have been many improvements such as its application to non-uniform costs setting in Field D* [5] or, more recently, Theta* [4, 6] which aims to remove the restriction on direction changes that generates A*.

The main difference between A* and Theta* is that the former only allows that the parent of a node is its predecessor, while in the last, the parent of a node can be any node. This property allows Theta* to find shorter paths with fewer turns compared to A*. However, this improvement implies a higher computational cost due to additional operations to be performed in the expansion nodes process. It is worth mentioning that these algorithms work on fully observable environments except Field D*, that can face partially observable environments applying a replanning scheme.

The Smooth Theta* (S-Theta*) [7] algorithm developed from Theta*, aims to reduce the amount of turns that the robot should perform to reach the goal. The motivation is that the robots consume more energy turning than going straight and rovers in other planets do not have fully rotability so they cannot rotate big angles. Then, we have modified Theta* to consider first in the search the expansion of nodes close to the current heading of the robot. The evaluation function of A* or Theta* is modified adding a third parameter, named α(p) that evaluates the direction of the successor node with respect to the current parent's position and the goal node.

But these algorithms work over a flat plane without valleys or hills. What made them not suitable for real environments such as the Martian surface.

3. SURFACE PATH-PLANNING

In the literature is hard to find path-planning algorithms that are focused on the mobility over Digital Terrain Models (DTM). Some approaches are appearing in the last years [12, 13, 14, 15], but they have limited application or require a high computational effort to obtain a route. We are focused on obtaining a general path-planning algorithm for the mobility over a realistic surface model, using the DTMs obtained by satellites. To do this, we have identified three important issues to achieve:

- The cost of a movement is a function of the distance between two points in a three-dimensional environment. A good model of the environment is required to compute distances with precision and apply some constraints, such as the maximum tilt of the rover.
- Use of transversal costs. They are defined as a terrain property that can be considered as a lineal function of a set of factors related to the mobility over a particular region of the map. This allows the algorithm to overcome potentially dangerous areas, such as quicksand.
- Evaluation of heading changes during search. Just like the S-Theta* algorithm [7], in the vertexes expansion process, the algorithm calculates the necessary turns needed to reach the next vertex taking into consideration the current heading and the position of the goal. This has two advantages: on the one hand, it could be used to obtain smoother routes, and on the other hand, there is the possibility to avoid routes with abrupt heading changes, which is appropriate for robots with limited turn capabilities .

Some work in this direction is being developed. Currently we have a modified version of S-Theta* that works with DTMs to provide safer routes between two points, considering heading changes, traversal costs and distances in 3D. Although some parameters can be used to adapt the path-planning algorithm to different necessities, some work is required to include safety constraint such as the maximum tilt allowed by the rover or modes to compute a good prediction of the battery consumption during the route.

4. UP2TA PLANNER: AN INTEGRATED SYSTEM

The idea behind the UP2TA planner (Unified Path-Planning and Task-planning Architecture) comes from our experiences using a PDDL planner as the deliberative layer for the control of an exploration robot [1]. In this way, we want to obtain from the planner a close to optimal ordering of multiple scientific tasks placed in a grid, and the paths to follow between them.

The problem of using a PDDL-based planner to both integrate path-planning and task-planning comes from two sides. First, the complexity of the codification of the domain and the problem. It is specially significant in the problem, where we need to declare all locations and their connections, that is, what points are adjacent with each other. And second, the planner must manage a huge amount of information since there are a lot of variables to instantiate before performing the search. And, in our case, with eight possible moves, this problem grows exponentially with the number of locations. Also, these planners use domain independent heuristics that are not useful for the path-planning problem. Related to this, in our experiments with this kind of PDDL codifications we often obtain rare paths with zig-zag patterns when solving these kind of problems.

In our planetary exploration problem, it is important to be able to manage big grids (at least 500x500). In those cases, the search is intractable and any state of the art planner cannot solve it. Then, we have two options; split the problem in two: we first compute the route for each goal and then we plan, or instead, we integrate the path-planning search inside the AI planner.

In order to solve these problems and probe the concept of integrating a PDDL-planner for task-planning and path-planning, we have merged a well-known PDDL planner, FF, and a surface path-planning algorithm to work together in a coordinated way. The resultant system, called UP2TA planner, takes the goodnesses of FF –for task planning using PDDL problems and independent domain heuristic; and the ones of the path-planning algorithm –for obtaining better routes using a specific domain heuristic and with the possibility of employing a DTM. Thereby, the scope of each module of the UP2TA planner is easy to identify: the task-planner is responsible of computing the actions to perform the scientific tasks contained in the PDDL files, and the path-planner is the one in charge of calculating safe and short routes between locations.

FF [8] as an Heuristic Search Planner transforms planning problems into problems of heuristic search by automatically extracting heuristics functions (*h*) from the problem, instead of introducing them manually [9]. Considering a relaxed problem in which all delete lists are ignored, from any state *s*, the optimal cost *h(s)* for solving the relaxed problem can be shown to be a lower bound on the optimal cost *h*(s)* for solving the original problem. As a result, the heuristic function *h(s)* can be used as an admissible heuristic for solving the original problem.

Since we do not want that the task planner has any path-planning related operation, the PDDL domain and problem must be redefined, and the information of the map can be adapted to the path-planning algorithm necessities. The files required for UP2TA are the following:

- *Domain and problem information:* it uses the same PDDL files shown in the previous section erasing the data for path-planning, that is, in the domain we remove the eight movement actions and predicates related to adjacent locations and obstacles. And, in the problem, we only need to define the locations in where we want to perform scientific tasks and the initial location of the robot.
- *Grid information:* we employ two (or more) files: one with the DTM information and at least another one with the position of the obstacles. Also, these files can contain the costs data related to move through each region of the map.

With these files, the UP2TA planner will provide a sequence of actions with the optimal (or close to optimal) distance among the tasks. Figure 1 shows the system.

The UP2TA planner works as a single system but it has two differentiate parts: the task-planner and the path-planner. The control of the systems resides in the task-planner. Since the task-planner is FF, the system behavior is nearly the same. The process starts with the lecture and

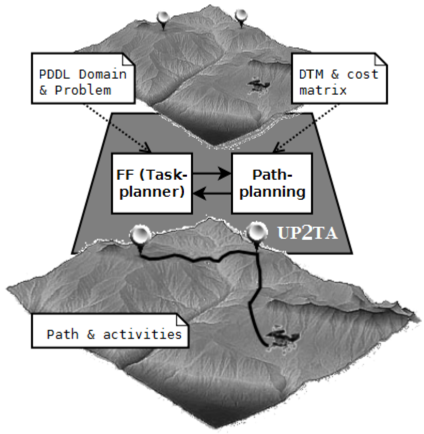

Figure 1. Representation of the UP2TA *expansion process for a problem with two tasks.*

instantiation of the PDDL data. That is, the robot's initial position and the scientific tasks to be performed. But we can introduce any other parameters or constraints into the PDDL model, such as the subsystems restrictions.

Then, the search begins: the UP2TA takes each task T_i and the current position ($start$) of the robot, and calculates its heuristic, both, the task-planner heuristic, $h_{FF}(\mathrm{T}_i)$ (given by a relaxed GRAPHPLAN [10]), and the one provided by the path-planning algorithm, $dist(start, \mathrm{T}_i)$. So we have merged these two heuristics to allow the system to decide the actions to plan first based on the distance.

We have modified the FF search in two ways: first we disable the Hill Climbing search to avoid local minimums; and then, we have transformed the Best-First Search algorithm into A*. To do this, we have added the $G(\mathrm{T}_i)$ value and we have expanded the nodes using the resultant $F(\mathrm{T}_i)$ value. $G(\mathrm{T}_i)$ represents the path cost from the current position to the location of the T_i task plus the cost to reach the current position from the start location.

At this point, we need to compute the path between the current location and all the other locations with reachable scientific tasks, and this is time consuming since the path-planning algorithm has to spend time in searching all of those possible routes. The solution adopted is to use a greedy path-planning algorithm [11] to calculate $G(\mathrm{T}_i)$. The idea behind these algorithms is to take the straight-line distance between the start and the goal position and try to follow this line. Nodes far from this line will not be expanded, and therefore, less memory and time is spent during the search of paths for partial plans (that is, a sequence of movements from one location to another location in which the robot must perform a scientific task T_n). Although this approximation does not obtain the real path, the associated path length allows us to take into consideration the presence of obstacles without spending a lot of time in obtaining paths between tasks that probably will not be used in the solution. We de-

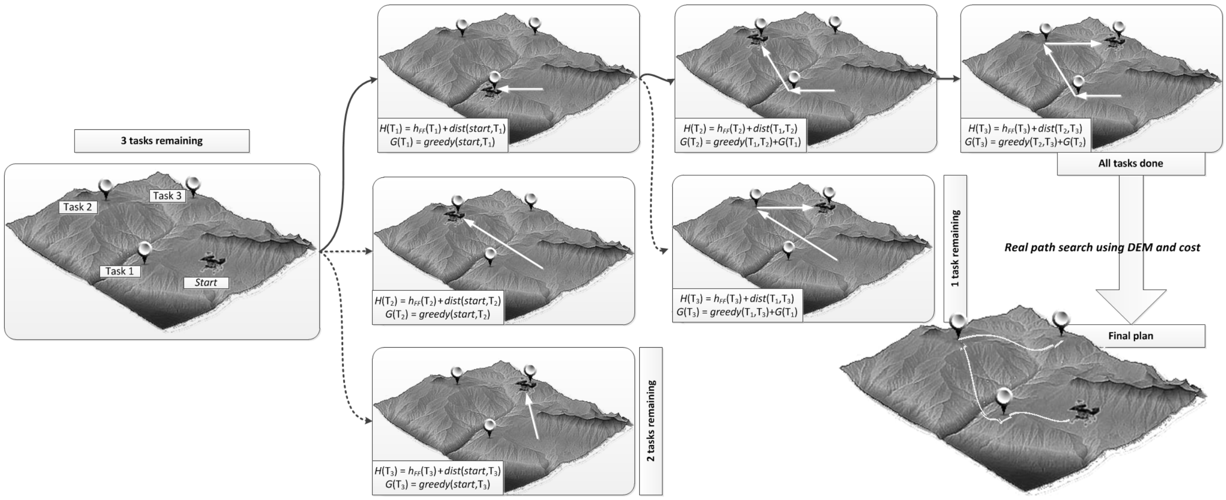

Figure 2. Example of UP2TA *search process with a problem of 3 tasks.*

fine $greedy(\mathrm{T}_i, \mathrm{T}_j)$ as the path cost between tasks T_i and T_j using a greedy path-planning algorithm. That greedy path length is added to the $G(\mathrm{T}_i)$ value of the partial plan and then the UP2TA planner adds all partial plans to a list, ordered by the $F(\mathrm{T}_i)$ values.

The search process continues with the extraction of the best promising partial plan. The UP2TA generates the possible partial plans starting from the new location as shown in fig. 2. In the same way as A*, the search process continues until there are no more plans, that is, there is no solution (unreachable task/s) or all tasks have been planned.

Figure 2 shows an example for the search process of the UP2TA planner with a problem of three tasks. The rover is located at the $start$ position.The first step is to calculate the heuristic and cost value for each task: $G(\mathrm{T}_i)$. The heuristic corresponds to the Euclidean distance between a task T_i and the current position plus the heuristic given by the task planner, in that case, the number of incomplete tasks. The cost of achieving the position of the task is calculated using the greedy path-planning algorithm as we have previously explained. Then there are three partial plans, one for each tasks, to be evaluated.

UP2TA takes the one with less $F(\mathrm{T}_i)$ value, in our example is the partial plan corresponding to perform the Task1 (T_1). Now it evaluates the other two possible tasks left. In the same way, the system calculates the heuristic and cost to reach Task2 and Task3 from Task1. Considering that Task2, for example, $G(\mathrm{T}_2)$ is the path length of the greedy path-planning algorithm between the tasks, $greedy(\mathrm{T}_1, \mathrm{T}_2)$, adding the cost to achieve the current task, $G(\mathrm{T}_1)$. In the example the best promising partial plan (i.e. shortest distance) is the one that first performs T_1 and then T_2. The best promising partial plans are connected through a bold line, while the others use a dotted line.

At this point there is only one task remaining, so the search finishes adding that task. Since all tasks are planned, the ordered tasks sequence for the solution is $\mathrm{T}_1 \rightarrow \mathrm{T}_2 \rightarrow \mathrm{T}_3$. Then, UP2TA will calculate the real path between each pair of consecutive tasks, that have been previously ordered. This path takes into consideration the altitude information of the terrain and the transversal cost that defines the terrain. In this way, the path length must be calculated taking into consideration the altitude difference between points, and thus, the path usually avoids big changes in the altitude. Also, our system implements the possibility of using one file that can represent the energy associated to move through every region of the map or the aridity of the terrain. That information allows us to model areas with high energy consumption or the hazard level in order to try to keep away, if possible, certain areas of the map.

Finally, the system provides the ordered list of tasks with the paths between them.

5. EXPERIMENTS ON MARS MAPS

In this section we show some examples of what we are currently developing for path-planning over DTMs. To test initial versions of our algorithm, we have used DTMs of Mars obtained by two NASA's orbiters: Mars Global Surveyor (MGS) and Mars Reconnaissance Observer (MRO). For the first one, we can obtain maps for the whole Mars surface with a maximum elevation resolution of 30 cm and a spatial resolution of 0.46305 km/px using the Mars Orbiter Laser Altimeter (MOLA)[1]. On the other hand, the High Resolution Imaging Science Experiment (HiRISE) instrument on board the MRO gives a vertical precision of 25 cm with a spatial resolution between 1 and 2 meters. But only a few DTMs are available[2] due

[1]Data available at http://www.mapaplanet.org/explorer/help/data set.html#mars mgs mola topo

[2]Available DTMs are at http://hirise.lpl.arizona.edu/dtm

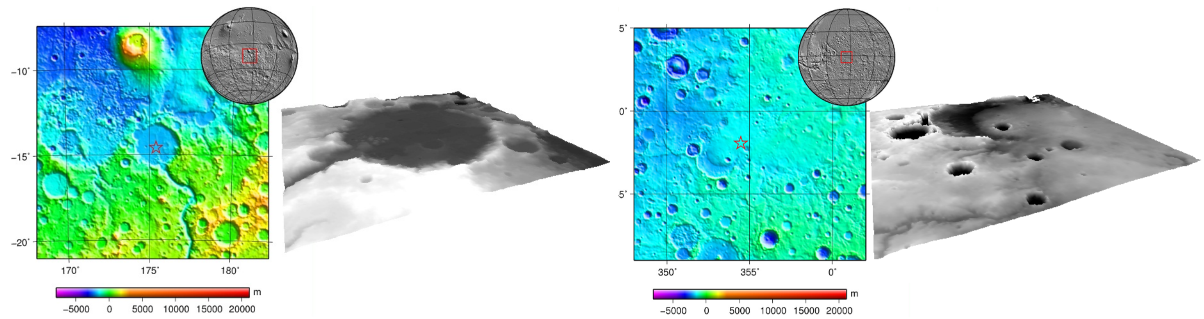

Figure 3. Landing site of Spirit rover (left) and Opportunity (right) with its 3D map reconstructions.

to the high computational effort required to generate the models.

We have taken DTMs from the landing site of the Mars Exploration Rover (MER) Spirit and Opportunity, using the MOLA instrument. Employed maps have 1184x1184 nodes with a real dimension of 236.8 km^2. Also, we have used an area of the landing site of the Mars Science Laboratory (MSL) as a part of the HiRISE catalog. The DTM employed has a spatial resolution of 1.01 m/px, and a dimension close to 17000x7000 points. Currently, we cannot deal with this map size, so we have only considered regions of 1000x1000 nodes, which approximately corresponds to an area of 1 km^2. Figure 3 shows both landing sites for Spirit and Opportunity rovers with the 3D map reconstructions.

The following figures (4, 5, 6 and 7) show the routes obtained by our planner. These maps represents the DTM model (more altitude is darker areas), and, instead of typical path-planning maps for testbench, in these maps there are no obstacles. We have taken representative start and goal points to test the behavior of the algorithm. For example, in fig. 4 we can see how the route follows a channel to overcome the side of the crater. Or fig. 7 in which the path-planning algorithm surrounds the crater.

As a small testbench we have generated 100 synthetic DTMs using the Hill algorithm [16]. These maps have 500x500 nodes and the elevation is in the range of 0–250. The execution has been performed in a basic PC taking the initial node on the coordinate $(0, 0)$ and the goal on the coordinates $(499, 450 - 499)$. Table 1 shows the result of applying our modified S-Theta* algorithm and a version of A* that takes into consideration the elevation of the nodes. We can see that our algorithm improves both, the path length and the total turns (the measurement of the heading changes during the route) respect to the modified A*. The number of vertex expanded does not differ too much, however the runtime is greater for the modified S-Theta*. This is mainly due to two reasons: one is the lack of optimization of the algorithm and the other one is the computational cost added by the interpolation method employed to obtain the elevation of a point between nodes. Also, this version includes the heading changes consideration explained by Muñoz and R-Moreno (2012) to smooth the path during the search.

Parameter	Mod. A*	Mod. S-Theta*
Path cost	854.94	817.70
Total turn	4014.90	112.80
Runtime (ms)	17644.68	72532.72
Vertex expansions	84841.90	94595.22

Table 1. Data from synthetic maps.

6. CONCLUSIONS

Planetary exploration missions will require new systems able to achieve the goals they were designed for. In some cases these capabilities can be met by adapting or further developing existing technology, but in other cases new innovative technology should be injected in order to exploit science opportunities and reduce overall operation costs. In this paper, we have focused on planning capabilities: the paths to follow and the scientific tasks to perform. Then, our goal is to try to optimally (or close to optimal) plan the paths that the planetary mission should follow having in mind the terrain features (slope, roughness, etc) and efficiently manage the energy consumption while performing the scientific tasks. We have presented a new system named UP2TA that is able to integrate path-planning and task planning capabilites. This new planner is able to provide safer routes between two points, considering heading changes, traversal costs and distances in 3D. Experiments have been performed on the landing sites of the Mars Exploration Rover (MER) Spirit and Opportunity.

ACKNOWLEDGMENTS

Pablo Muñoz is supported by the European Space Agency (ESA) under the Networking and Partnering Initiative (NPI) *Cooperative systems for autonomous exploration missions*. This work was partially supported by the Spanish CDTI project COLSUVH, leaded by the *Ixion Industry and Aerospace* company.

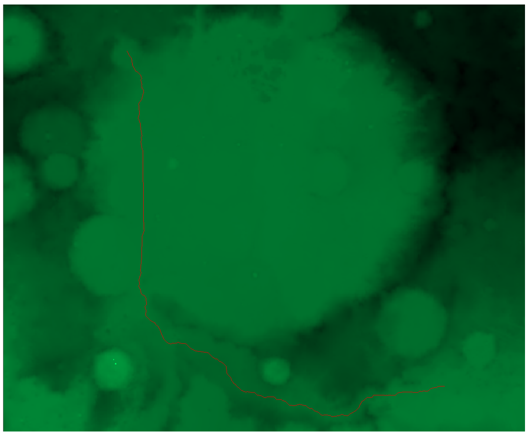

Figure 4. Representation of a route considering the DTM model of Spirit landing site.

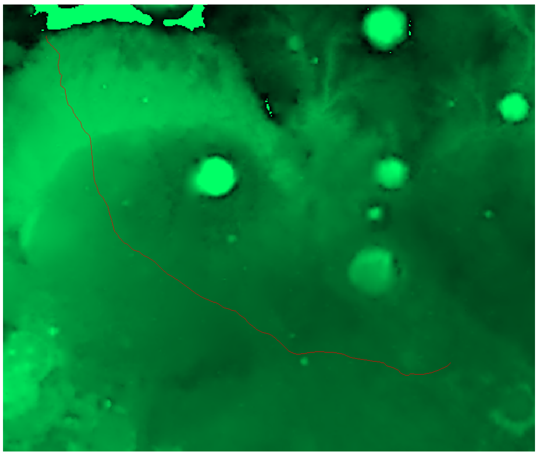

Figure 5. Representation of a route considering the DTM model of Opportunity landing site.

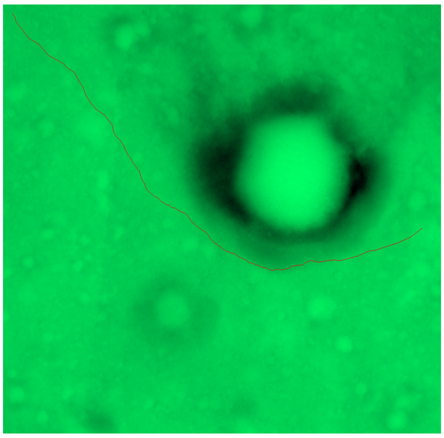

Figure 6. Representation of a route outside the Gale crater on Mars using part of a high resolution DTM.

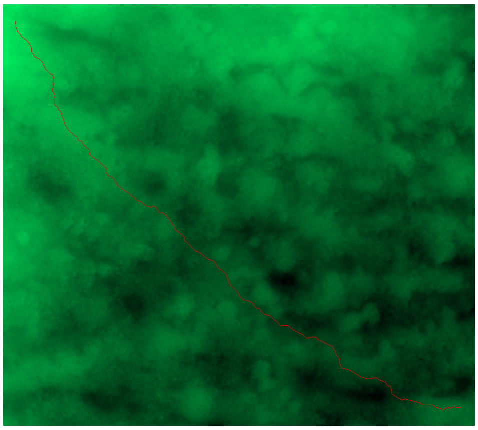

Figure 7. Another region of the Gale crater on Mars solved with the modified Theta.*

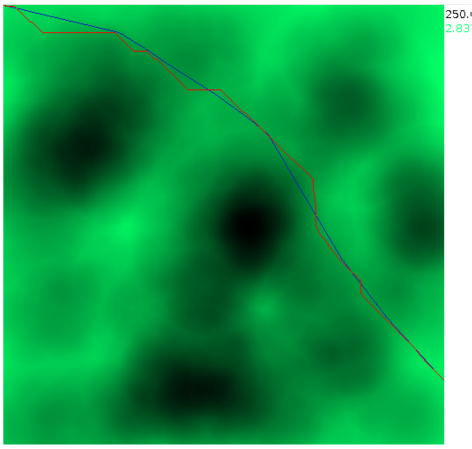

Figure 8. Synthetic map solved with modified A (red) and S-Theta* (blue).*

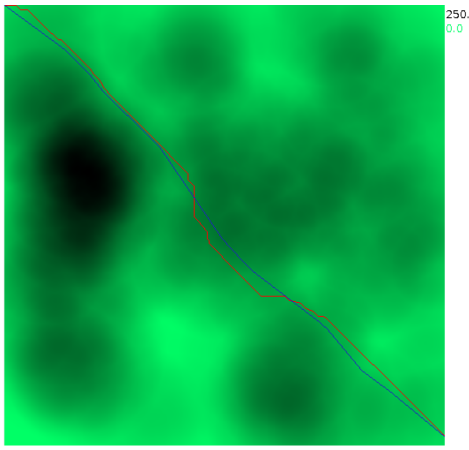

Figure 9. Another synthetic map solved with modified A (red) and S-Theta* (blue).*

REFERENCES

[1] Muñoz, P. and R-Moreno and M. D. and Martínez, A. A first approach for the autonomy of the Exomars rover using a 3-Tier architecture, In Procs. of the 11th ESA Workshop on Advanced Space Technologies for Robotics and Automation. Noordwijk, The Netherlands, 2011.

[2] Yap, P. Grid-Based Path-Finding, Advances in Artificial Intelligence, 2338: 44-55, Lecture Notes in Computer Science, Springer Berlin, 2002.

[3] Hart, P. and Nilsson, N.J. and Raphael, B. A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics, 4:100-107, 1968.

[4] Nash, A. and Daniel, K. and Koenig, S. and Felner, A. Theta*: Any-Angle Path Planning on Grids, In Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI'07), pp: 1177-1183, Vancouver, British Columbia, Canada, 2007.

[5] Ferguson, D. and Stentz, A. Field D*: An Interpolation-based Path Planner and Replanner, In Proceedings of the 12th International Symposium on Robotics Research (ISRR), San Francisco, CA, USA, 2005.

[6] Daniel, K. and Nash. A. and Koening, S. and Felner, A. Theta*: Any-Angle Path Planning on Grids, Journal of Artificial Intelligence Research, Vol. 39, pp. 533-579, 2010.

[7] Muñoz, P. and R-Moreno, M. D. S-Theta*: low steering path-planning algorithm, In Procs. of the 32nd SGAI International Conference on Artificial Intelligence, Cambridge, UK, 2012.

[8] Hoffmann, J, and Nebel, B. The FF Planning System: Fast Plan Generation Through Heuristic Search, Artificial Intelligence Research, 14:253-302, 2001.

[9] Bonet, B. and Geffner, H. Planning as Heuristic Search: New results, In Procs. of the European Conference on Planning (ECP-99), Durham, UK, 1999.

[10] blum, A., and Furst, M. Fast Planning Through Planning Graph Analysis. Artificial Intelligence, 90 (1997), 281?300.

[11] Muñoz, P. and R-Moreno, M. D. Improving Efficiency in Any-Angle Path-Planning Algorithms, In Procs. of the 6th IEEE International Conference on Intelligent Systems, Sofia, Bulgaria, 2012.

[12] D. L. Page, A. F. Koschan, M. A. Abidi and J. L. Overholt, Ridge-Valley Path Planning for 3D Terrains, in International Conference on Robotics and Automation, Orlando, Florida, May 2006.

[13] G. Ishigami, K. Nagatani and K. Yoshida, Path Planning for Planetary Exploration Rovers and Its Evaluation based on Wheel Slip Dynamics, in IEEE International Conference on Robotics and Automation (ICRA2007), Roma, Italy, April 2007.

[14] A. Garcia, A. Barrientos, A. Medina, P. Colmenarejo, L. Mollinedo and C. Rossi, 3D Path planning using a fuzzy logic navigational map for Planetary Surface Rovers, In Procs. of the 11th ESA Workshop on Advanced Space Technologies for Robotics and Automation. Noordwijk, The Netherlands, 2011.

[15] S. Choi, J. Park, E. Lim, and W. Yu, Global Path Planning on Uneven Elevation Maps, in Proceedings of the International Conference on Ubiquitous Robot and Ambient Intelligence (URAI), 2012.

[16] Douglas, D. and Peucker, T. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, The Canadian Cartographer, 10: 112-122, 1973.