JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 46, No. 4, July–August 2009

Landing Point Designation Algorithm for Lunar Landing

Babak E. Cohanim* and Brian K. Collins†
Charles Stark Draper Laboratory, Cambridge, Massachusetts 02139

DOI: 10.2514/1.42002

Safe, precise, and efficient lunar landings must detect and avoid hazardous boulders, craters, and slopes while identifying predeployed assets and managing fuel consumption. This paper describes an algorithm to designate landing points during lunar descent. Inputs to the algorithm are the lander footprint, divert ΔV costs, detected hazards, and identified points of interest. The algorithm is designed to accept any number of input cost functions and constraints, ensuring flexibility to a variety of missions. The input cost functions are weighted and input into a search algorithm that returns all feasible landing points on the map, prioritized by the weighted cost function. The output of the algorithm is a set of landing aim points. The algorithm provides onboard crew or autonomous robotic vehicles with landing point designation capabilities.

Nomenclature

DTNH	=	distance to nearest hazard map, m
DTNPOI	=	distance to nearest point of interest map, m
n	=	side length of map, pixesl
spread	=	overlap of solutions, pixels
VFDE	=	vehicle footprint dispersion ellipse defined as the lander footprint + guidance, navigation, control errors at touchdown, m
ΔV	=	change in velocity, mps

I. Introduction

NASA'S Vision for Space Exploration (VSE) [1] outlines the return of manned missions to the moon and Mars in the upcoming decades. The key factors for landing crew and cargo safely on the lunar or Martian surface are detecting and avoiding hazards during descent and landing, landing close to predeployed assets, and minimizing the ΔV of divert maneuvers. This paper describes an algorithm that designates landing aim points (LAPs) for a vehicle during planetary descent based on the factors that affect the safety, precision, and efficiency of the mission.

To ensure safety, future missions will need to detect and avoid hazards at the landing site. Past missions, like Apollo, performed hazard detection and avoidance without the aid of hazard detection and avoidance sensors and algorithms [2]. Premission planning assessed orbital imagery and determined the areas that had the least amount of hazardous terrain. Apollo landings were near-equatorial, upon relatively hazard free terrain, and under ideal lighting conditions. The Apollo trajectories were designed to accommodate crew viewing out the Lunar Excursion Module window, with ideal lighting conditions. This allowed the crew to assess the lunar terrain upon descent, pick a suitable hazard free area, and divert to that area for landing.

NASA's Constellation program plans to return to the moon to test new exploration technologies and establish a stepping stone for missions to Mars and beyond. Robotic missions identified craters near the lunar poles as locations which may harbor water ice. These polar locations are good candidates for future exploration and lunar outposts [3]. The poles have significant challenges in terrain and lighting conditions over Apollo era landings [4]. Low sun angles cast long shadows and mask hazards. The terrain is more rugged, including larger craters and boulders in greater frequency. Lunar Reconnaissance Orbiter (LRO) imagery will provide adequate identification of the general slopes, large boulders, and large craters on the lunar surface, as well as major points of interest such as predeployed surface assets, scientific areas, and lit areas [5]. Local slopes, shallow craters, and small boulders that are below the resolution of orbital reconnaissance may cause the vehicle to tip. To ensure safe landings and provide onboard crew and autonomous systems with the necessary situational awareness, active detection of hazards and predeployed assets will be needed during descent. Descent vehicles will also estimate fuel levels and an onboard algorithm will provide ΔV estimates for possible diverts as the vehicle descends to the landing site. Active detection and fuel estimation are especially crucial for missions to new sites, at which safe landing locations may be far from the a priori landing target.

These three key pieces of information, hazards, points of interest, and ΔV, can be used to quickly assess landing aim points on the surface by producing a weighted cost function of the "distances to nearest hazards" (DTNH), "distances to nearest points of interest" (DTNPOI), and ΔV contours. The weighted cost function is a two-dimensional surface cost map that will be used by the designation algorithm to choose landing aim points that are safe, close to areas of interest, and minimize diverts.

This paper discusses possible weighting schemes between the objectives, but assumes that before the missions, extensive research will be performed to determine the actual weighting scheme desired for each mission. This paper will focus on the algorithm description for producing alternate landing aim points based on two pieces of information: the weighted cost function and the effective lander footprint.

This algorithm is being used as part of the development of the Autonomous Landing and Hazard Avoidance Technology (ALHAT) Project out of NASA's Exploration Technology Development Program Office (ETDPO). The algorithm was developed to use hazard detection algorithms designed for use on Mars [6]. The Constellation Architecture Requirements Document (CARD) [7] and ALHAT operations concept and requirements [8] describe the requirements relative to predefined landing targets that were used in defining the hazard tolerances and scan areas used for the analysis presented in this paper. The algorithm also interacts with the human interactive capability of the Autonomous Flight Manager (AFM) in the ALHAT System [9] to provide the crew with situational awareness on the location of hazards, points of interest, and divert capabilities while allowing the crew to designate landing aim points.

II. Input Parameters

The algorithm is designed to optimize several different types of maps of the surface to determine candidate landing aim points. For

Presented as Paper 6818 at the AIAA GNC 2008, Honolulu, HI, 17–22 August 2008; received 4 November 2008; revision received 20 April 2009; accepted for publication 13 April 2009. Copyright © 2009 by Charles Stark Draper Laboratory. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Senior Member of the Technical Staff, 555 Technology Sq, MS 27; bcohanim@draper.com

†Senior Member of the Technical Staff, 555 Technology Sq, MS 15; bcollins@draper.com

this paper, three input cost maps have been identified that represent performance, cost, and risk.

At the landing site, there may be predeployed assets, scientific areas, or desired lighting conditions which require vehicles to land precisely near, but not on, these points of interest. The "performance" metric chosen to maximize that ability is to land as close to given assets as possible. The CARD [7] has placed precision landing requirements on lunar sortie and outpost missions. For outpost buildup, having outpost elements land as close as possible to each other minimizes the moving or cabling that might be needed to hook up separately landed units, but dust and plume may limit how close landers can get to predeployed elements. For sortie missions, at which plume contamination or dust obscuration may not be as much of an issue, there may be several areas of interest in a given landing site. These can be prioritized based on the desires of mission planners and can be recorded in a two-dimensional cost map. The cost map gives lower costs to different points of interest on the ground with increasing contours emanating out from the points of interest. Figure 1 shows an example DTNPOI cost map. The "points of interest" in this map are located at a cost of zero (black). Contours are created from zero (black) to one (white) based on the distance from the point of interest. Points of interest can be prioritized by giving the most desired point a value of zero and subsequent points of interest higher values.

The maps presented in this paper are $n \times n$ m^2 in area. The size of these maps is determined by analysis showing that there is a high probability that at least one safe site will be located within the landing area. Johnson et al. [10] analyzed safe map sizes as a function of vehicle hazard tolerance and the vehicle footprint dispersion ellipse (VFDE). The VFDE is the lander footprint with guidance, navigation, and control (GN&C) errors expected at touchdown added as a buffer. A hazard is defined as any local slope, rock, hole, or crater that may cause the vehicle to tip, or damage the landing gear. Future reconnaissance and analysis of the proposed lunar landing sites will determine the size of the landing areas required to ensure a high probability of finding at least one safe site at the landing area.

Large-scale hazards, such as highly sloped regions, large craters, and large boulders, are expected to be identified well before any lander mission to the moon is launched by either LRO [5] or Earth-based radar measurements [11]. But smaller craters or boulders, which can still damage the lander, may not be detected until the landing site comes in view during the descent phase. Regardless of when and how the hazards are detected, a hazard map, which is the "risk" metric used in this paper, will need to be created to effectively perform hazard avoidance during landing. During Apollo missions, hazard detection and avoidance was handled by the crew looking out the window, under ideal lighting conditions and into limited hazard fields [2]. Missions proposed in the new Constellation Architecture may go to areas that are highly sloped, cratered, and/or boulder strewn [3]. The crew may no longer be able to perform hazard detection and avoidance visually, but will need the assistance of onboard hazard maps and algorithms to determine safe landing aim points. The hazard cost map will consist of "keep-out" zones. The keep-out zones are the areas around each hazard, or groups of hazards, where the vehicle is not allowed to land. This is defined as the distance that is equal to or less than the radius of the VFDE in any transverse direction from the hazard. Figure 2 shows a representative diagram of the VFDE. Small ellipses represent the lander footpads (outer ellipses) and engine bell (inner ellipse), each 1 m in diameter. Medium-sized ellipses encompassing the footpads are a 1 m additional error allotment around the footpads. The large ellipse encompassing all footpads and the engine bell is the VFDE, approximately 10 m in radius for the analysis presented in this paper.

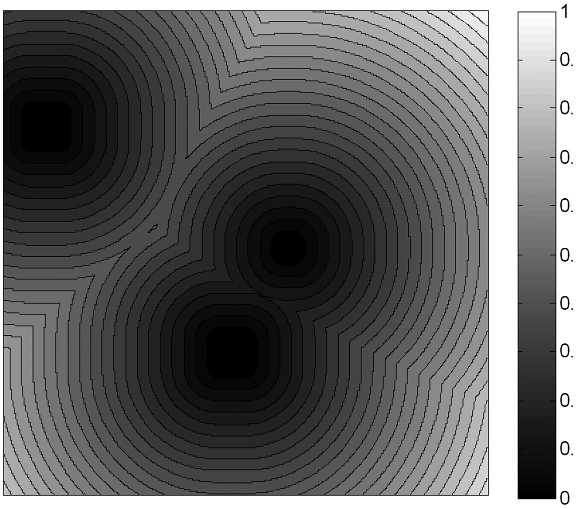

Fig. 1 DTNPOI cost map.

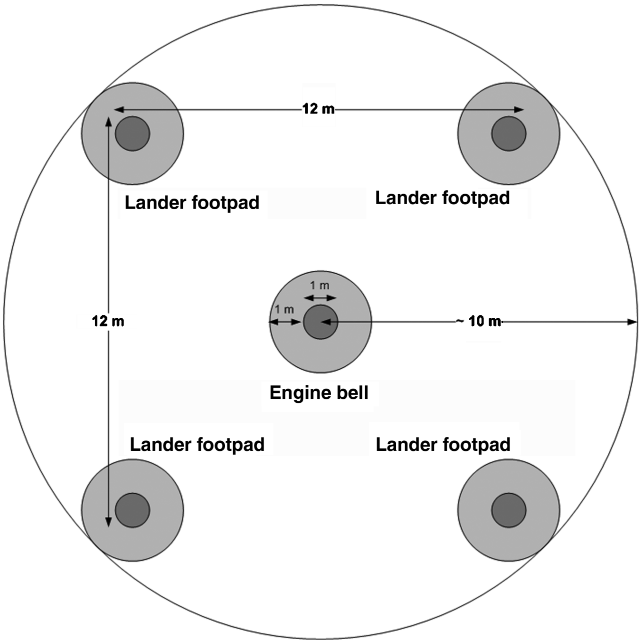

Fig. 2 Vehicle footprint dispersion ellipse (VFDE).

There may be a desire to have the vehicle straddle hazards in the open volume between the lander footpads and the engine bell, which is possible, but may not be practical for a number of reasons. Safety margins added on top of the VFDE could diminish any benefit of defining the footpads separately. Landing with the footpad VFDE configuration may require an undesirable roll maneuver near touchdown which could disorient the crew and increase the expected GN&C errors. The safest and most robust method is to define the VFDE as an ellipse that defines the span of diagonal footpads, plus a ring around which defines the GN&C errors.

The DTNH cost map can be defined similarly to the DTNPOI map, but the cost of hazards are the inverse of the performance benefit of points of interest. The DTNPOI cost map has the lowest cost at the points of interest and increased cost with increasing distance from these points. Hazards and the accompanying areas around them within the lander radius have prohibitively high costs (equal to one), and so the algorithm will not choose to land the vehicle in the "keep-out" zone. Points that are outside of a VFDE radius from hazards are considered safe and costs decrease as the distance from hazards increase. Points that are furthest from hazards have the lowest cost (equal to zero). There may be a point defined by mission planners for which "far enough is good enough" and essentially the cost of being away from a hazard is essentially even, for example, being 20 m away from any hazard is just as safe as being 30 m away from any hazard. This can be easily implemented in creation of the DTNH cost map by mission planners by defining the distance that is "far enough." This paper assumes that the further away from a hazard, the better.

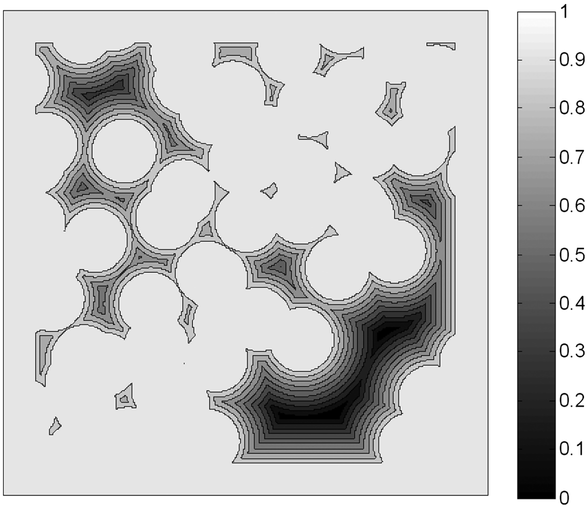

Fig. 3 DTNH cost map.

Figure 3 shows an example DTNH map that has hazardous regions (equal to one) in the brightest white, while points that are furthest from any hazard (equal to zero) are darkest black.

The landing target is set into memory and the descent guidance is trying to hit that point as its final target. The ΔV contour map is assumed to be centered on the original landing target, that is, the minimum fuel consumption point is located at the point of the landing target, which defines the landing site and is thus centered in the cost map. Divert ΔV is the "cost" metric. Any divert away from the original target will cost ΔV and thus fuel. Small diverts cost small amounts of ΔV and still leave plenty of margin in case another redirection must occur. Larger diverts cost more ΔV and leave less margin because they begin to impinge on the edge of the landing site and thus closer to the undefined hazard zone outside the edge of the hazard map. If a second redirection is needed, even more ΔV may be needed to possibly go all the way back across the landing site to another safe site. Figure 4 shows an example of the ΔV map. The point at zero (darkest black) represents the minimal amount of additional ΔV needed for divert while points close to one (white) have the highest relative costs.

The weighted cost function is created from the three cost maps mentioned previously: DTNH map, DTNPOI map, and ΔV map. The three cost maps are normalized in two ways to make it easier to compare them to one another. First, as mentioned previously, the maps are all of the same area. The information in each map is interpolated, if necessary, to create maps that are the same dimensions, and thus contain the same number of pixels. Second, the values for DTNH, DTNPOI, and ΔV are normalized between zero and one. This allows each pixel in each map to have the same potential cost. Weighting between the various cost maps can now be calculated automatically in one of two ways: equal weighting or "dynamic" weighting. Equal weighting gives each pixel the same weight from map to map, so when they are weighted against each other the final weighted cost function will equally represent each cost map value at that coordinate. Dynamic weighting adjusts the weight of each pixel in each map by the relative cost of one cost map to the others. Normalization is key for dynamic weighting. Because each map is the same size, that is, contains the same number of pixels, and all pixels are normalized between zero and one, each map has the same potential maximum cost equal to $n \times n$ pixels. For example, if a given cost map has 100×100 pixels, the potential maximum cost of that map would be 10,000 if all pixels in that map were at the highest cost of one. Conversely if each pixel was at the lowest possible cost of zero, the cost of the map would be zero, and if each pixel were 0.5 in cost, the total map cost would be 5000. This simple sum can be used to measure the relative cost of each map to another. Maps that have higher total costs can be seen as more stressing to the solution, and can then be weighted higher. For example, if there are many hazards in a given area, the DTNH cost map will have many points that are equal to or near one, as opposed to an area that has few hazards and only has points equal to or near one in a small portion of the map. The sum of each cost map can be easily computed and divided by the sum of all pixels in all maps to compute the relative weight of each map. The following equations are for equal and dynamic weighting, showing how each weight is calculated. The sum of the weights will add up to one.

Equal:

$$w_1 = w_2 = w_3 = \ldots = w_N = \frac{1}{N} \tag{1}$$

Dynamic:

$$\text{MapCost}_i = \sum_{1}^{M\times M} \text{Value}_{M\times M} \tag{2}$$

$$w_i = \frac{\text{Mapcost}_i}{\sum_{j=1:\,N} \text{Mapcost}_j} \tag{3}$$

where N is the number of cost maps and M is the number of pixels on a side of each map.

The weighting may also be assigned a priori by mission planners and may be adjusted by the crew during flight. Either way, the algorithm is independent from the weighting scheme. As long as there is a two-dimensional cost function that captures the

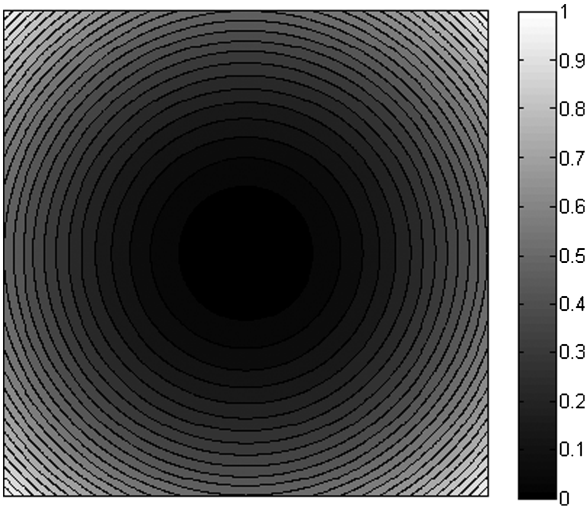

Fig. 4 ΔV cost map.

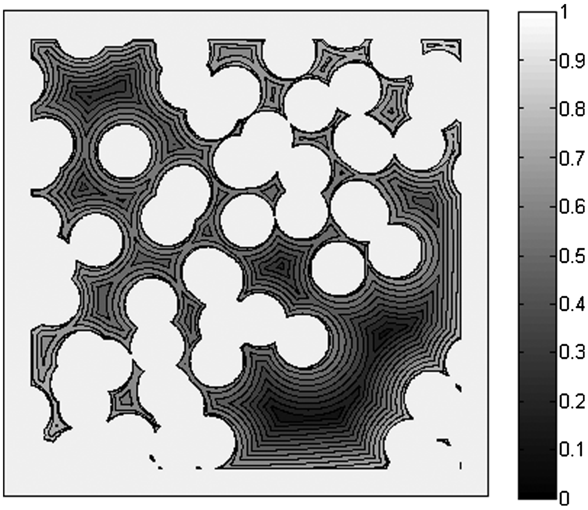

Fig. 5 Weighted cost map using dynamic weighting.

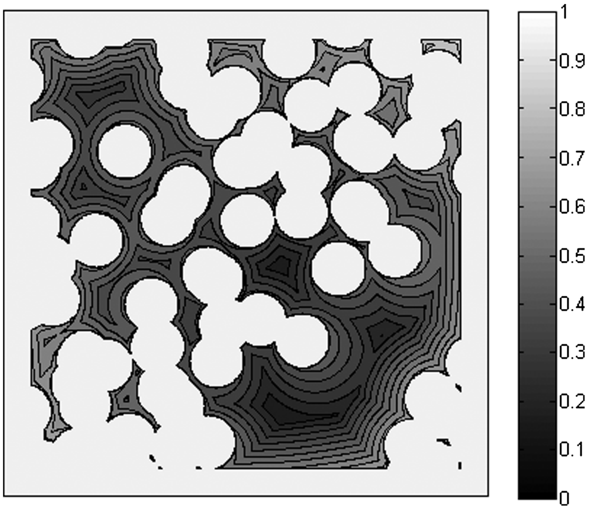

Fig. 6 Weighted cost map using equal weighting.

performance, cost, and risk of the landing point, the algorithm will find landing aim points.

There are special considerations that need to be discussed in terms of the weighted cost function: the hazards on the map, and the vehicle footprint. It is assumed that hazards create total keep-out zones on which no part of the lander footprint can overlap. So, in the creation of the weighted cost function, at any place in which there is a hazard within the radius of the VFDE, the maximum cost is applied to ensure that the vehicle will avoid the hazard and land safely. Figure 5 shows the weighted cost map using dynamic weighting for the input cost maps in Figs. 1, 3, and 4. Points with a lower cost (blacker) are valued over points with higher costs (whiter). Infeasible portions of the map are given a value of one (brightest white) and denote regions that are unsafe for landing. Figure 6 shows the weighted cost map using equal weighting using the same cost maps.

III. Algorithm

The goal of the algorithm is to return landing aim points based on the two-dimensional weighted cost function. The algorithm is a greedy algorithm that finds the lowest-cost solution first, and then subsequent solutions from there on out based on the desired “spread” of results. The spread is a parameter that determines how much chosen solutions are allowed to overlap by defining the distance from the center of one solution to the center of another. This can be as close as one pixel to just less than the diameter of the VFDE (overlapping solutions), to the diameter of the VFDE (solutions touch, but do not overlap), or to a distance that is greater than the VFDE diameter (additional buffer between solutions). A natural choice for the spread is equal to or slightly greater than the diameter of the lander footprint so as to create nonoverlapping, distinct landing aim points. The spread can be adjusted to create overlapping landing aim points or provide additional buffer between points. The spread can be adjusted by operators or the crew to produce varying landing aim points. Figure 7 shows the three spread scenarios. Figure 7a shows a spread that is less than the VFDE diameter (overlapping). Figure 7b is a spread that equals the VFDE diameter (touching, but not overlapping). Figure 7c shows a spread that is greater than the VFDE diameter (additional buffer).

The algorithm begins by sorting all points by their weighted cost function value. The point with the lowest cost is chosen and placed into a “bin” designating a candidate landing aim point. The point with the next lowest value is then pulled from the sorted list and a comparison is made to determine if the spread from the already binned lowest value point overlaps with the value of the point currently under consideration. If not, the point is considered a viable option and also placed into the “bin.” If so, then the point is dismissed, and the next lowest point is chosen in a greedy fashion. This continues until either all points are evaluated, points below a certain cost are all evaluated, or a certain number of points have been recorded (the algorithm is flexible in this regard). Points under current consideration are only compared against points that are already placed in the “bin.” Because dismissed points have already been compared against points in the bin and have been deemed dismissed, there need not be wasted computation in comparing them with the point under current consideration. If the spread is set in such a way that nearly full overlap is allowed (points within a map pixel are allowed), then all points are viable bin options and one would only need to sort the list based on values. Figure 8 shows a flow diagram of this process.

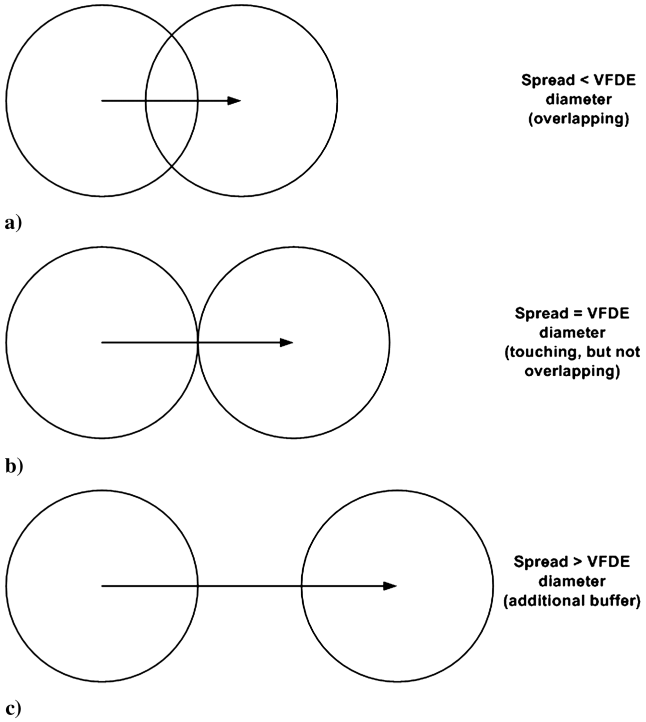
Spread < VFDE diameter (overlapping)
a)
Spread = VFDE diameter (touching, but not overlapping)
b)
Spread > VFDE diameter (additional buffer)
c)

Fig. 7 Spread scenarios: a) Spread < VFDE diameter (overlapping); b) Spread = VFDE diameter (touching, but not overlapping); c) Spread > VFDE diameter (additional buffer).

IV. Output Parameters

The algorithm returns coordinates, in the local two-dimensional map frame of reference, of candidate landing aim points based on the weighted cost function. The candidate landing aim points can be used for further post processing by the crew to aid in decision making. For autonomous vehicles, the landing aim points can be transformed into the guidance system’s operating frame and used directly to guide the vehicle to the best landing aim point. Figure 9 shows the weighted cost function with hazards and landing aim points using dynamic weighting and spread equal to the lander diameter. Bright white represents hazardous areas. Small black dots are hazards. Shaded contour regions are feasible areas that could contain landing aim points. The prioritized landing aim points are white ellipses representing the VFDE (to scale) with the rank given as the number located in the center of the ellipse. One can see from the figure that a region the radius of the VFDE around a hazard is at a cost of one (white).

Figure 10 uses the same cost maps and spread, but with equal weighting. As can be seen by the differences in the LAPs displayed between the two plots, differences in weighting cause differences in the final solution, but the number one landing aim point remains the same.

Figure 11 shows the solution for dynamic weighting, but the spread parameter is set to 2× the VFDE diameter. As can be seen from the differences between Figs. 9 and 11, increasing the spread has created solutions that are further apart and provide solutions in new areas of the map.

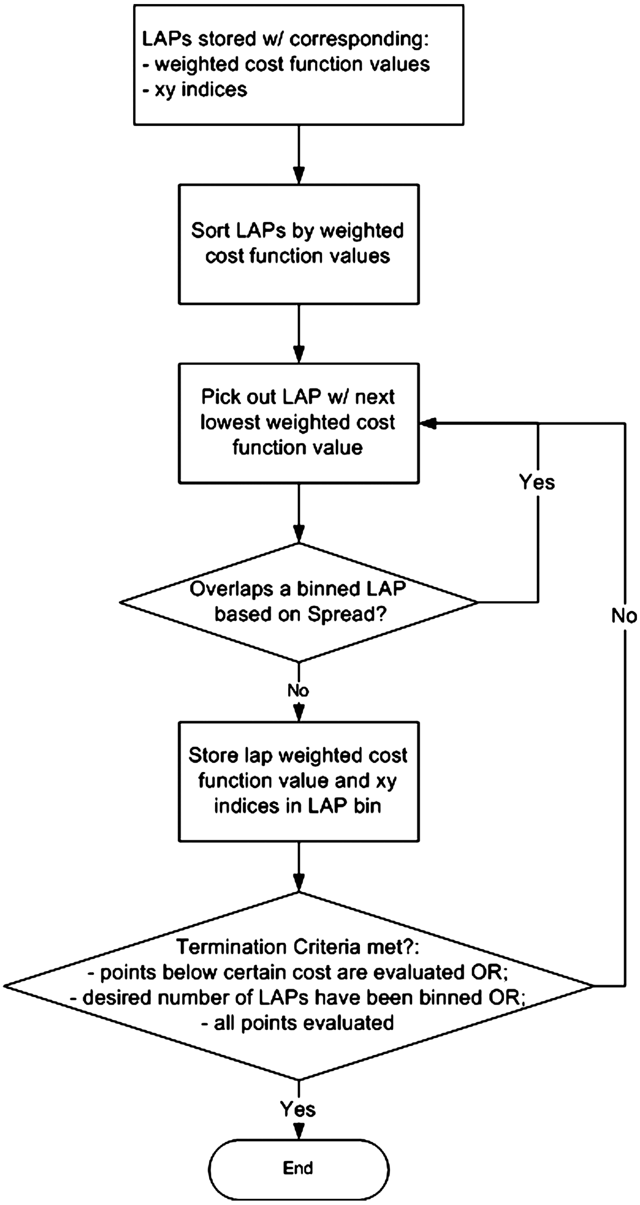
LAPs stored w/ corresponding:
- weighted cost function values
- xy indices
Sort LAPs by weighted cost function values
Pick out LAP w/ next lowest weighted cost function value
Overlaps a binned LAP based on Spread?
Yes
No
No
Store lap weighted cost function value and xy indices in LAP bin
Termination Criteria met?:
- points below certain cost are evaluated OR;
- desired number of LAPs have been binned OR;
- all points evaluated
Yes
End

Fig. 8 Flow diagram of candidate landing aim points selection.

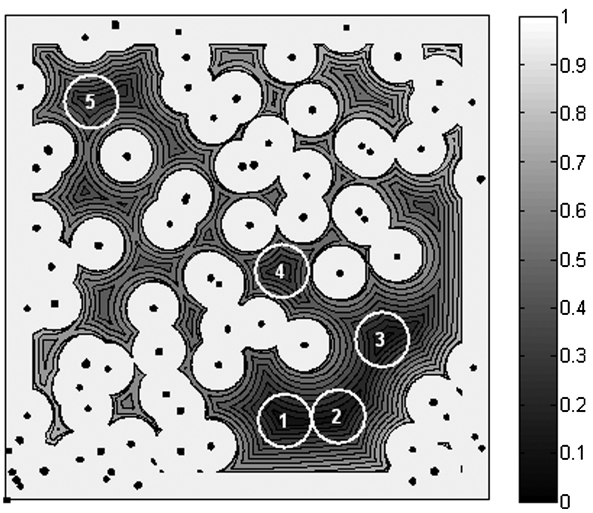

Fig. 9 Solutions for dynamic weighting and Spread = VFDE diameter.

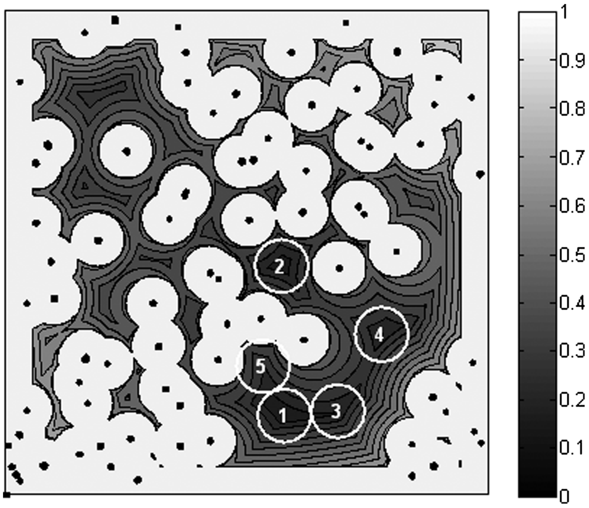

Fig. 10 Solutions for equal weighting and Spread = VFDE diameter.

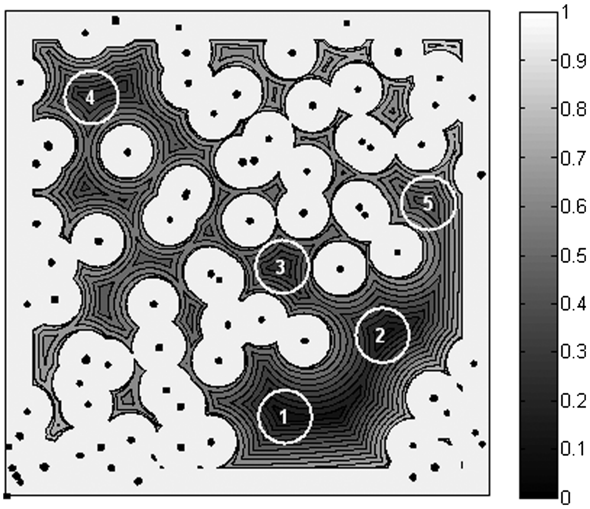

Fig. 11 Solutions for dynamic weighting and Spread = 2× the VFDE diameter.

Figure 12 shows the solution for dynamic weighting, but the spread parameter is set to 0.667× the VFDE diameter. As can be seen from the differences between Figs. 9 and 12, decreasing the spread has created solutions that are clumped closer together in an area of the map that has a large safe area.

The algorithm has added flexibility to quickly output multiple landing aim point sets based on various spreads, which can be assessed without recomputing the weighted cost function. The spread process is applied after the weighted cost function has been computed and the LAPs have been sorted, so there is no need to recalculate the weighted cost function or re-sort. This is useful for providing speed and flexibility for crew-in-the-loop testing of the algorithm. The algorithm is also flexible enough to return only the answers that showed up as local minima from the weighted cost function.

V. Discussion

A. Subtleties in the Algorithm

The purpose of the algorithm is to find points in the landing site that are a safe distance from hazards, close to points of interest, and close to the minimum ΔV usage. Because this is an optimization process, interesting phenomena may occur at the place in which all the generated landing aim points are clumped into a single area. The algorithm has been developed and tested to produce what are referred

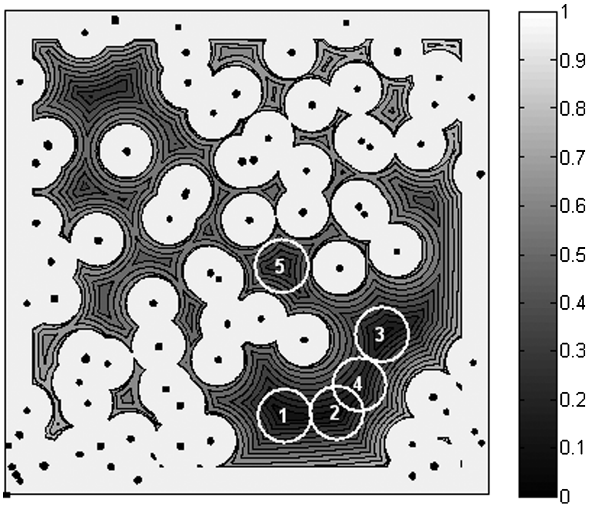

Fig. 12 Solutions for dynamic weighting and Spread = 0.667× the VFDE diameter.

to as "distinct" solutions in the map. An example of some extreme cases may shed better light on this topic. Imagine a perfectly benign surface that contains no hazards (flat slope, no rocks, no craters) and no particular points of interest, for example, a first-down landing on a surface that is perfectly flat within the landing site. This is highly unlikely, but if it does occur the interaction of the three cost maps will produce a weighted cost function that reports a single landing aim point in the center of the map. Exploring this further will show why this occurs: beyond the edge of the landing site map it is assumed that hazard information is untrustworthy, so the edge of the map is considered "hazardous." The DTNH map would appear as in Fig. 13, the DTNPOI map would be totally flat (all costs zero), and the ΔV map would be biased toward the middle of the landing site as shown in Fig. 4. The weighting will thus create one distinct area that is focused toward the center of the landing site. There are no anisotropies to create distinction in the map. This does not mean that there is only one landing aim point in the map, of course, but demonstrates an interesting phenomena that is mathematically possible. Figure 13 shows the DTNH cost map with no hazards in the field. The point on the map which is the furthest from any hazard is at a cost of zero (black). In this case only the edges of the map are considered hazardous because anything beyond those boundaries have not been detected and thus are unknown.

B. Trading Off DTNH from DTNPOI

There may be instances when a hazard is also an area of interest, for example, a crater is hazardous to land in, but may harbor ice deposits or other interesting science. The tradeoff between DTNH and DTNPOI will push the algorithm toward landing as close to the hazard/area of interest as possible while still keeping the lander a safe distance away. An example of this can be graphically represented in Figs. 14 and 15 in which there are hazardous craters that are also defined as areas of interest. The goal is to keep the lander out of the hazardous areas, and the algorithm's chosen landing aim point places the lander close to the craters. In Fig. 14a, the surface digital elevation map (DEM) shows the contours of an example lunar surface with three craters located in the field (the color bar shows elevation in m). The ΔV cost map is focused toward the center of the map. The DTNH cost map shows areas that are far from hazards (black) versus areas that overlap a hazard (white). The DTNPOI cost map shows the three craters as centers of points of interest. The DTNH and DTNPOI cost maps are nearly reversed in cost, but the DTNH cost map does have values that are equal to one, which will force the algorithm to avoid placing a landing aim point in those regions.

Figure 15 shows the weighted cost function that evenly weighs ΔV, DTNH, and DTNPOI cost maps from Fig. 14. The prioritized landing aim points are white ellipses representing the VFDE (to scale) with the rank given as the number located in the center of the ellipse. Hazardous areas are solid black surrounded by white denoting the keep-out zones.

C. Ties in the Weighted Cost Function

Regardless of the spread that is chosen, the landing aim point with the lowest cost is always chosen as the first landing aim point

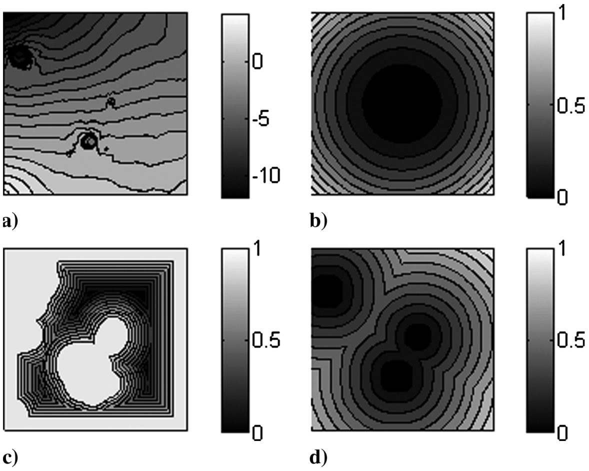

Fig. 14 DTNH vs DTNPOI example: a) Surface DEM; b) ΔV; c) DTNH; d) DTNPOI.

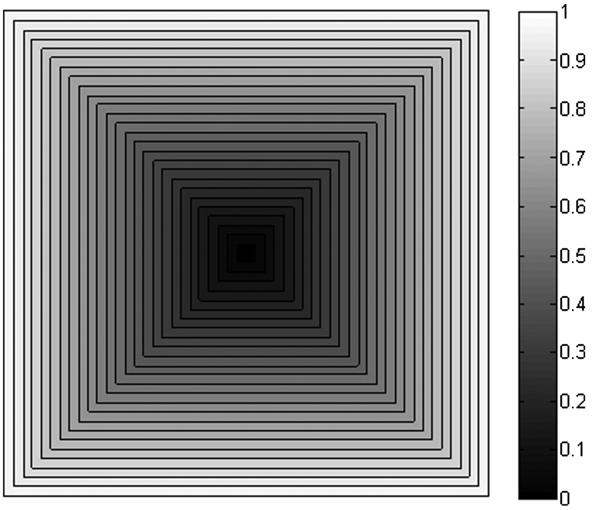

Fig. 13 DTNH cost map with no hazards in the field.

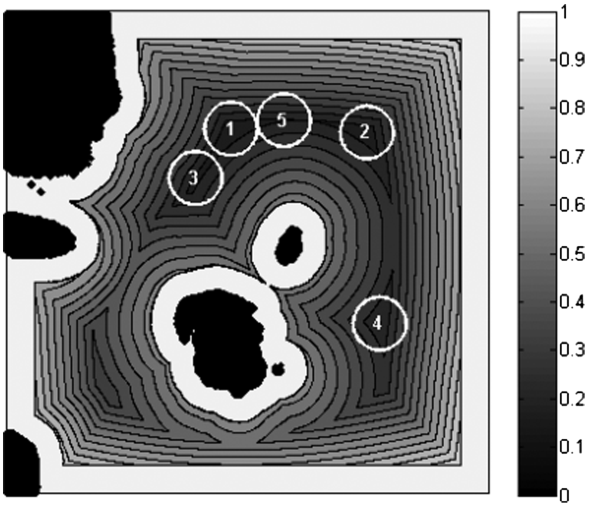

Fig. 15 Dynamic weighted cost function for DTNH vs DTNPOI.

returned as a solution. This point is the global minimum in the weighted cost function. Because of the naturally occurring parabolic tendencies in the ΔV cost map, it is highly improbable that two points will ever have the exact same value. This only occurs in the “keep-out zones” in which the points are set to the maximum value and should be duly ignored. Nonetheless, to reduce the probability even further, a very, very small random offset can be subtracted from the cost of each point in the weighted cost function that is under the maximum cost, reducing the chances down to the precision of the offset (which can be set to the last significant digit of the implemented precision of the algorithm). Another method of alleviating ties can be to recognize if a tie is occurring between neighboring points and picking the one with the lowest average cost in a neighboring area around each point. Depending on the application and use of the algorithm, ties can be reported as ties, or can be randomly prioritized.

VI. Conclusions

We have created a landing point designation algorithm that provides a method to enable safe and precise planetary landings for crewed or robotic vehicles. The algorithm searches a two-dimensional cost map to enable an efficient assessment during planetary descent, at which detailed hazard data may become fully available for the first time. Distance to nearest hazard, distance to points of interest, and ΔV contours were used as examples of cost maps that could be used with this algorithm, but any cost map can be created and used in the assessment of the landing site providing flexibility to a large set of mission types. The implementation of a spread parameter allows flexibility to access landing aim points at various spacing in the given landing site. Coupled with an onboard hazard detection system, a priori reconnaissance of hazards and areas of interest, and algorithms that estimate divert ΔV, our algorithm provides a powerful method for enabling safe, precise, and efficient designation of a landing point during planetary descent.

Acknowledgments

The authors would like to thank Tye Brady, Laura Major, Lauren Kessler, Steve Paschall, and Doug Zimpfer of Draper Labs for contributions to the ideas in this paper. The authors would also like to thank Andrew Johnson of Jet Propulsion Laboratory for the numerous DEMs and hazard data provided for testing. This work was completed under contract NNJ06HC37C for NASA’s ETDPO under the ALHAT Project.

References

[1] “The Vision for Space Exploration,” NASA, Feb. 2004.

[2] Bennett, F. V., “Apollo Experience Report—Mission Planning for Lunar Module Descent and Ascent,” NASA Rept. D-6846, June 1972.

[3] Connolly, J. F., “Constellation Program Overview,” NASA Constellation Program Office, Oct. 2006.

[4] Brady, T., Schwartz, J., and Tillier, C., “System Architecture and Operation Concept for an Autonomous Precision Lunar Landing System,” *Proceedings of the 30th Annual American Astronautical Society Guidance and Control Conference*, AAS, Paper 07-053, Feb. 2007.

[5] Keller, J., and Chin, G., “Lunar Reconnaissance Orbiter: Instrument Suite and Objectives,” Goddard Space Flight Center, IAC-07-A3.6.A.03.

[6] Johnson, A. E., Klumpp, A. R., Collier, J. B., and Wolf, A. A., “Lidar-Based Hazard Avoidance for Safe Landing on Mars,” *Journal of Guidance, Control, and Dynamics*, , Vol. 25, No. 6, 2002, pp. 1091–1099.
doi:10.2514/2.4988

[7] “Constellation Architecture Requirements Document (CARD),” NASA CxP 70000, Revision B, Feb. 2008.

[8] Epp, C. D., Robertson, E. A., and Brady, T., “Autonomous Landing and Hazard Avoidance Technology (ALHAT),” *Proceedings of the IEEE/AIAA Aerospace 2008 Conference*, IEEE, Paper 1644, 2008.

[9] Forest, L. M., Cohanim, B. E., and Brady, T., “Human Interactive Landing Point Redesignation for Lunar Landing,” *Proceedings of the IEEE/AIAA Aerospace 2008 Conference*, IEEE, Paper 1642, 2008.

[10] Johnson, A. E., Huertas, A., Werner, R. A., and Montgomery, J. F., “Analysis of On-Board Hazard Detection and Avoidance for Safe Lunar Landing,” *Proceedings of the IEEE/AIAA Aerospace 2008 Conference*, IEEE, Paper 1656, 2008.

[11] Campbell, B. A., Campbell, D. B., Chandler, J. F., Hine, A. A., Nolan, M. C., and Perillat, P. J., “Radar Imaging of the Lunar Poles,” *Nature (London)*, Vol. 426, , 2003, pp. 137–138.
doi:10.1038/426137a

C. Kluever
Associate Editor