

Roughness Map for Autonomous Rovers

Jonathan Edau Loh¹, Gabriel Hugh Elkaim², Renwick E. Curry³

Abstract—All past NASA planetary rovers have only been able to traverse exo-planetary surfaces at a maximum speed of $0.05 - 0.09m/s$. Greater autonomy and higher speed (up to $3m/s$) would increase their utility and extend the range of coverage. A novel roughness metric is used to create a roughness map and provide planetary rovers with qualitative information about the terrain and difficulty of traversal. The method is deployed in real time on NASA’s K-Rex rover and shows excellent capability for distinguishing passable from impassable terrain. This roughness metric shows superior discrimination to other established methods of obstacle avoidance and terrain classification. It is computationally efficient, and runs in linear time. Experimental results on the NASA ARC MarsScape favorably compare and contrast our roughness metric to other existing methods.

I. INTRODUCTION

Over the past 40 years, all exo-planetary rovers have been tele-operated: a human controller commanding the rover to *drive forward 20 turns* or to *drive to this location*. Since the Mars Pathfinder (1996), exo-planetary rovers have all included some level of autonomy implemented in the form of hazard avoidance systems to navigate through unknown terrain [1]. The Mars Exploration Rover Spirit (2004) was the first exo-planetary rover to achieve autonomous driving and path planning to target destinations between $40 - 50m$ away [2][3]. Within the past decade, research in autonomous navigation and path planning has focused on rover traversability over rough terrain [4], [5], [6].

There is a continual effort to increase rover top speed and to predict a safe speed over rough terrain to minimize vibration effects on sensors, and avoid crashing or damaging on-board electronics. In order to traverse terrain faster, it is important to locate roughness as areas to avoid. In autonomous mobile robotics, traversing through unknown terrain has generally been approached using an obstacle detection algorithm [7], wheel contact forces [8], and roughness cost functions [5], [9].

In this paper we present a novel method that creates a roughness map providing planetary rovers with local and global information about the terrain. The algorithm for the roughness map represents the terrain from incoming sensor data and uses this information to calculate surface roughnesses. The surface roughness provides a hazard/obstacle

detection for the terrain with the goal to drive slowly over rough terrain (i.e.: rocks, ditches, peaks) and faster over relatively flat terrain. This roughness map provides better path planning information and will also aid the control operator by illuminating areas of hazards even while in tele-operation mode.

II. RELATED WORK

Research defining terrain roughness is a rich and well established area [8], [7]. Previous planetary rovers such as the Mars Exploration Rovers (MER) and the Mars Science Laboratory (MSL), have used cameras and stereo vision to detect obstacles/hazards and categorize terrain as traversable or not traversable [10], [11]. The subject rover, KRex, is a LiDAR based rover that uses Probabilistic Terrain Analysis (PTA), which is a fast and lightweight obstacle detection algorithm developed in [7]. PTA is very useful for detecting obstacles to avoid, however it is a binary detection algorithm providing a simple go/no-go solution.

III. BACKGROUND

A. KRex System

The system used for experimental validation is NASA ARC’s KRex rover (pictured in Fig. 1). These algorithms have also been tested on NASA JSC’s Centaur2. KRex is a four-wheel drive all-wheel steering rover with a top speed of $3[m/s]$. It is $4.22m^3$, and has a modular mast with sensors attached. The KRex rover uses a Velodyne



Fig. 1: NASA ARC KRex

LiDAR to detect path conditions and obstacles; it has 32 beams that provide simultaneous range measurements from

¹J.E. Loh is with the Department of Computer Engineering, University of California Santa Cruz, Santa Cruz, CA, USA jonathan.loh@gmail.com

²G.H. Elkaim is with the Faculty of the Department of Computer Engineering, University of California Santa Cruz, Santa Cruz, CA, USA elkaim@soe.ucsc.edu

³R. E. Curry is with the Faculty of the Department of Computer Engineering, University of California Santa Cruz, Santa Cruz, CA, USA rcurry@ucsc.edu

-30° to +10° from horizontal body axis, with a complete 360° scan occurring at 10Hz[12]. The Velodyne sensor has a range of 70 meters and produces some 700,000 points per second. When accumulated, these points form a 3-D point cloud which is registered to a global grid map using a pose estimate. After the point cloud data (PCD) has been updated with pose, the PCD is considered aligned. These pose updates are based upon compass, inclinometer, GPS, and Inertial Measurement Unit (IMU) measurements as well as matching points to their corresponding plane.

B. Grid Map

The environment is mapped as an array of cells ranging in size from 5cm to 50cm depending on the application requirements. We improve upon the PTA and use a grid to store accumulated statistics for each cell. Each cell of the grid contains estimates of surface roughness. The grid representation helps discretize the environment uniformly and organize the sensor data.

IV. ROUGHNESS MAP

A. Terrain Model Representation

The first step in creating the roughness map is to recast the point cloud data (PCD) into a form that will best model and represent the terrain. With over 700,000 points per second coming from the Velodyne, we store the statistics into a matrix for each cell of a given terrain; this integrates seamlessly into the grid map infrastructure. Fig. 2 is an example grid map used for the KRex system where the tile size is the discoverable region around the rover as it traverses through the environment.

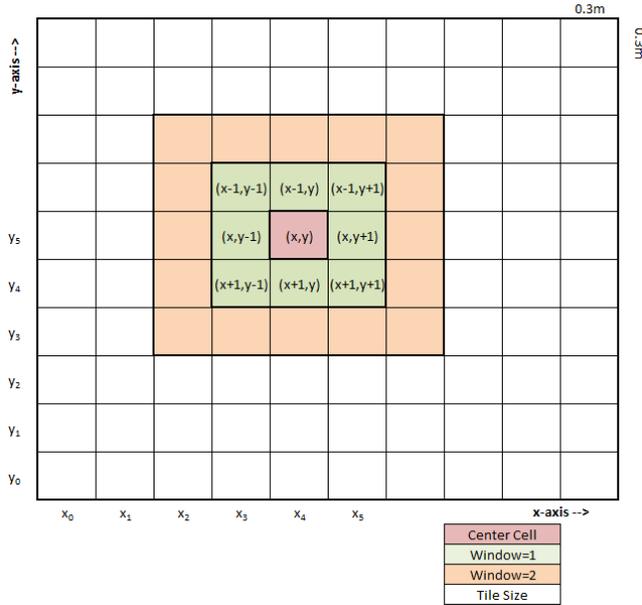


Fig. 2: Grid Map showing Tile Size, Cell Size, and Window Sizes where Window=1 is in green and includes the Center Cell; Window=2 is in orange and includes Window=1 and Center Cell

B. Fitting Plane Surfaces

A plane surface was fit to the PCD data in each cell using the following equation for a plane

$$z = Ax + By + C \quad (1)$$

This equation was used to estimate the vertical height of a point in a least squares fit. The statistics in the normal equation below are updated in real time as new/updated terrain data arrives.

$$\underbrace{\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & n \end{bmatrix}}_F \underbrace{\begin{bmatrix} A \\ B \\ C \end{bmatrix}}_x = \underbrace{\begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix}}_y \quad (2)$$

Once the best plane fit is determined, the surface normal is given by:

$$N = \frac{(-A, -B, 1)}{\|(-A, -B, 1)\|} \quad (3)$$

The least squares plane fit and surface normal give additional insight about surface direction. It also provides a contiguous relationship between neighboring cells when computing terrain smoothness (roughness).

C. Roughness Calculation

Roughness is defined as the compliment of the sum of products of smoothness cost functions ($s_{(\cdot)}$): $r = 1 - (\sum \Pi s_{(\cdot)})^\psi$.

1) *Surface Normals (N)*: From the surface normals (N), a smoothness scale (s_N) is created by taking the dot products of the unit normals from the center cell and its neighboring cells (Eq. 4) and applying an averaging filter over a given window size, K , where each side of the window has a length of $2K + 1$ cells (see Fig. 2).

$$s_N = |N_{(x,y)} \cdot N_{(x+i,y+j)}| \quad (4)$$

$$s(x,y) = \frac{1}{n_{cells}} \sum_{i=-K}^K \sum_{\substack{j=-K \\ (i,j) \neq (0,0)}}^K s_N \quad (5)$$

Where $n_{cells} = (2K + 1)^2 - 1$ is the number of cells inside a window size minus the center cell and (x, y) are Cartesian coordinates of a global grid map. The relationship focused upon is the θ from the dot product, $\frac{a \cdot b}{\|a\| \|b\|} = \cos\theta$. This tells us how much the neighboring cell planes are rotated relative to the center cell.

Using the dot product of the surface normals as a smoothness metric provides better intrinsic information about the terrain compared to [5] and [8].

2) *Within Cell Attributes*: This section examines three attributes of roughness within a cell: *residual* accounts for errors in the least squares fit; *coverage* accounts for the spread of the PCD points across the cell; *slope* incorporates the rovers pitch and roll limits.

a) *Residual*: The residual is the sum of the squares of the errors (or offsets) of the points from the plane (variance of plane errors). As the measurement is the z_i 's, the offset is in the vertical dimension. This is used to test the quality of the plane fit along the z-axis. This residual cost will be used as a weighting function in the total smoothness cost, using a logarithmic scale of the residual:

$$q = \frac{\log_{10}(\text{residual}^{-1})}{2.6021} \quad (6)$$

$$\text{residualCost} = \begin{cases} 0 & \text{if } q \leq 0 \\ 1 & \text{if } q \geq 1 \\ q & \text{else} \end{cases} \quad (7)$$

The scaling factor of $\frac{1}{2.6021} = \frac{1}{\log_{10}(0.05^{-2})}$ is used to put a lower limit on the residual and defines that residuals less than 5^2cm^2 indicate well fit planes.

b) *Coverage*: Coverage is how well the PCD is spread over the grid cell in the X-Y plane. Because the laser scan is a line scanner rotating 360 degrees, there is potential for the PCD to form a single line across the grid cell. If this were to occur, the plane fitting would be based upon the single line and would be a poor representation due to the lack of coverage.

In order to test the quality of the coverage, or planarity, the covariance method of principal component analysis (PCA) will be used. The eigenvalues are determined via the quadratic equation from the determinant of the covariance matrix C . Where C is a 2×2 matrix:

$$C(x, y) = \begin{bmatrix} \text{cov}(x, x) & \text{cov}(x, y) \\ \text{cov}(y, x) & \text{cov}(y, y) \end{bmatrix} \quad (8)$$

When testing for planarity, if either eigenvalue is less than or equal to ω then the PCD does not have good coverage of the grid cell. Where $(\lambda_i \geq 0)$ and $(0 \leq \omega < \text{cellSize})$.

$$\text{coverage} = \begin{cases} 0 & \text{if } \lambda_1 \leq \omega \text{ or } \lambda_2 \leq \omega \\ 1 & \text{else} \end{cases} \quad (9)$$

Experimentally, $\omega = 0.0008$ gave acceptable results as the coverage threshold, but we leave it to further research to determine what threshold for ω is considered bad coverage with respect to cellSize .

c) *Slope* (ϕ, θ): The slope is used to calculate traversability for singular cells. This incorporates the rover's roll (ϕ) and pitch (θ) limitations.

$$\text{slope} = \max \left[\frac{|A|}{\phi_{\text{thrs}}}, \frac{|B|}{\theta_{\text{thrs}}} \right] \quad (10)$$

where A and B are the plane coefficients, and ϕ_{thrs} and θ_{thrs} are the vehicle roll and pitch thresholds. We define the slope cost function as

$$s_{\text{slope}} = \begin{cases} 0 & \text{if } \text{slope} \geq 1 \\ 1 - \text{slope}^\eta & \text{else} \end{cases} \quad (11)$$

Experimentally we used $\eta = 4$ because it has an equivalent smoothness cost of 0.5 at 50° . η must be optimized to each specific mission.

3) *Projected Height* (μ): The surface normal metric has difficulty distinguishing a step in the terrain if the step happened to be perfectly aligned with the grid map. To handle this characteristic, we use a projected height of the plane (μ_{proj}) (see Fig. 3). Since the plane of each cell is centered at (0,0) and the cell size is known, finding the projected height is done by computing the height at the middle of the neighboring plane using the center plane equation:

$$\mu_{\text{proj}} = A_{\text{center}}(i * \text{cellSize}) + B_{\text{center}}(j * \text{cellSize}) + C_{\text{center}} \quad (12)$$

where i and j iterate through the neighboring cells. The neighboring cell height is found using the neighboring plane equation:

$$\mu_{\text{nbr}} = A_{\text{nbr}}(0) + B_{\text{nbr}}(0) + C_{\text{nbr}} \quad (13)$$

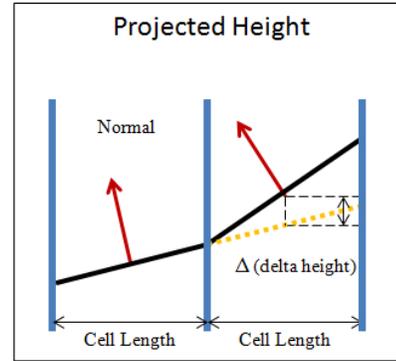


Fig. 3: Projected Height

The cost function is then:

$$\Delta = |\mu_{\text{proj}} - \mu_{\text{nbr}}| \quad (14)$$

$$s_{\mu}(x, y) = 1 - \min\left[\frac{\Delta}{\delta}, 1\right] \quad (15)$$

where δ is the vehicle obstacle height. Using projected height provides continuity between cells and checks if the terrain is a continuous slope or sloped step.

4) *Total Smoothness Cost*: The next step in the smoothness calculation is combining all the smoothness metrics together. To do this, a confidence factor (β) is introduced into the equation to help weigh the smoothness value between each cell more heavily for well-fit planes (low *residual*), good coverage (non-zero *eig*), and large cluster of points in cell (large *numPts*). Additionally, the center cell has its own confidence factor (γ) that will weigh the entire smoothness value more heavily for a well fit plane (*residualCost*), good *coverage*, and traversable slope (*s_{slope}*). This gives us our

updated smoothness equation $s_{center}(x, y)$:

$$s_{center}(x, y) = \frac{\gamma(x, y)}{\beta^*(x, y)} \sum_{i=-K}^K \sum_{\substack{j=-K \\ (i,j) \neq (0,0)}}^K (s_N * s_\mu * s_{slope} * \beta)(x + i, y + j) \quad (16)$$

$$\gamma(x, y) = residualCost(x, y) * coverage(x, y) * s_{slope}(x, y) \quad (17)$$

$$\beta^*(x, y) = \sum_{i=-K}^K \sum_{\substack{j=-K \\ (i,j) \neq (0,0)}}^K \beta(x + i, y + j) \quad (18)$$

$$\beta(x, y) = numPts(x, y) * residualCost(x, y) * coverage(x, y) \quad (19)$$

The inner term, $s_N * s_\mu * s_{slope}$, is computed to guarantee any smoothness cost equal to 0 ($s_N = 0$, $s_\mu = 0$, or $s_{slope} = 0$) will force the entire product to be rough. If $residualCost(x, y) = 0$ or $s_{slope}(x, y) = 0$, the terrain is untraversable due to bumpy terrain or slope exceeding the roll/pitch threshold. If $coverage(x, y) = 0$, the terrain is considered untraversable due to a lack of data quality and not actual terrain roughness. The data quality is included because for uncertain terrain we do not want the rover to traverse these areas.

Once the smoothness cost is computed for all the cells within the tile frame, the smoothness is smoothed using the neighboring smoothness' via the α term which is bounded from $[\frac{1}{n+1} : 1]$ where n is the number of neighboring cells used.

$$s(x, y) = \alpha * s_{center}(x, y) + (1 - \alpha) * s_{nbr}(x, y) \quad (20)$$

$$s_{nbr}(x, y) = \frac{1}{n} \sum_{i=-K}^K \sum_{\substack{j=-K \\ (i,j) \neq (0,0)}}^K s_{center}(x + i, y + j) \quad (21)$$

At the upper limit of $\alpha = 1$, the resulting smoothness ($s(x, y)$) will only contain the center cell smoothness (s_{center}). The lower limit of α is $\frac{1}{n+1}$, which provides equal weighting among all the $(n + 1)$ cells used within the window size including the center cell. The maximum number of neighboring cells within the window size is $n = (2K + 1)^2 - 1$.

a) *Smoothness Scaling*: It is safe to assume terrain areas with low smoothness (large roughness) are impassable and therefore these regions are penalized without adversely affecting obstacle representation. To accomplish this, the smoothness s is modified:

$$s'(x, y) = s(x, y)^\psi \quad (22)$$

Where $\psi > 1$. Raising the power of s greater than 1 maintains the range $[0, 1]$ while increasingly penalizing terrain regions that are rough. This concept is taken from [8]. Experimentally, $\psi = 2$ produced good results. Again, ψ must be optimized for the mission requirements.

5) *Final Roughness Cost*: The final roughness cost (r) is defined as the compliment of smoothness s' , so that a

roughness of 1 means rough, and a roughness of 0 means smooth.

$$r(x, y) = 1 - s'(x, y) \quad (23)$$

V. EXPERIMENTAL RESULTS

A. Roughness Map Method Results

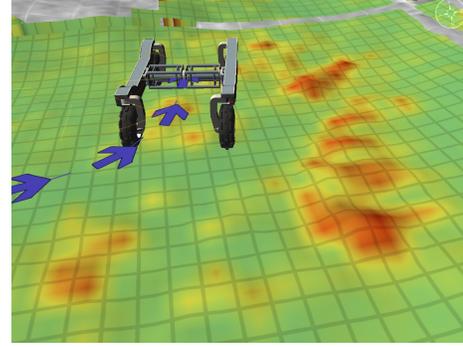


Fig. 4: Marscape - uneven terrain patch in red, showing dot product technique

1) *Dot Product Normal Results*: The effectiveness of the dot product normal in the roughness map method is seen in Fig. 4 of an uneven terrain patch on Marscape, where the bumps are less than $0.1725m$ high. These patches are considered rough because the dot product normal (s_N) is more prevalent in detecting changes in direction of the uneven surfaces than the slopes of the cells (s_{slope}) or projected heights (s_μ).

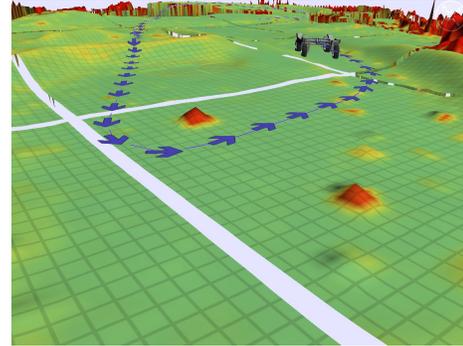


Fig. 5: Marscape - two large rocks $0.4572m$ tall, showing step terrain

2) *Obstacle Height Results*: We tested the obstacle height ($\delta = 0.3m$) using two singular large rocks, approximately $0.4572m$ tall and $0.60m$ wide, as seen in Fig. 5 by the red colored cells. All three smoothness cost functions (s_N, s_μ, s_{slope}) contributed to the roughness of the rocks because 1) the surface normals between rock cells have a large angle of rotation, 2) the projected height cost (s_μ) within the rocks exceeded δ , and 3) the rock slopes are greater than the roll/pitch thresholds. The two rocks have been computed as rough obstacles, which is noted by the red coloring of the rocks.

3) *Roll/Pitch Threshold Results:* We tested the slope smoothness (s_{slope}) using the crater like area of Marscape (see Fig. 6). Since craters are naturally continuously sloped, the majority of the terrain is computed as smooth, as evidenced by the green coloring, because the majority of the surface normals between crater cells have a small angle of rotation and the s_{μ} within the crater are near 0. Several areas near the apex of the crater have been marked rough (orange-yellow coloring) due to slopes approaching roll/pitch threshold and curvature changes of the terrain.

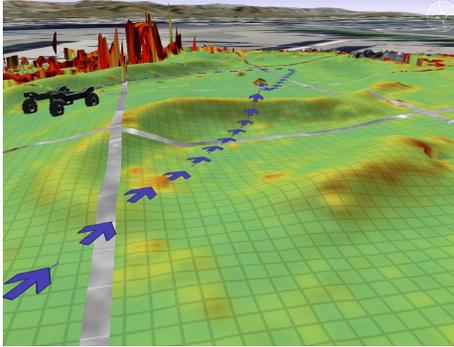


Fig. 6: Marscape - crater, showing continuous sloped terrain

B. DEM vs. LiDAR

We compare the Digital Elevation Map (DEM) of the NASA Ames Marscape with a $10cm$ resolution to the results from the KRex LiDAR system, using the same terrain area. This comparison is done in order to verify and validate the KRex system is detecting surface roughnesses similar to the DEM results.

The roughness map algorithm was applied to the Marscape DEM in Matlab and the results are shown in Fig. 7. The color gradient used— [dark blue→red] for [rough→ smooth].

The numbered boxes in Fig. 7, are terrain features that will be used to verify the KRex LiDAR system results. The terrain features are categorized as follows:

Feature	Description
1	Rough hillside
2	Uneven terrain on on level ground
3	Summit of hill
4	Crater wall
5	2 large rocks standing $0.4572m$ tall

TABLE I: Category of Terrain Features

The real-time results of the roughness map algorithm using the PCD from the LiDAR scans were simulated in VERVE, and the results of this are shown in Fig. 8. The color gradient of the speed map is [red→ green] which is equivalent to [rough → smooth].

The results of the DEM and LiDAR scan are nearly identical, which validates the consistency of the algorithm. Boxes 1-5 in Fig. 8 show features detected on the KRex system match the true data of the DEM in Fig. 7. The box 5 features are not shown well in Fig. 8 but is captured in Fig. 5.

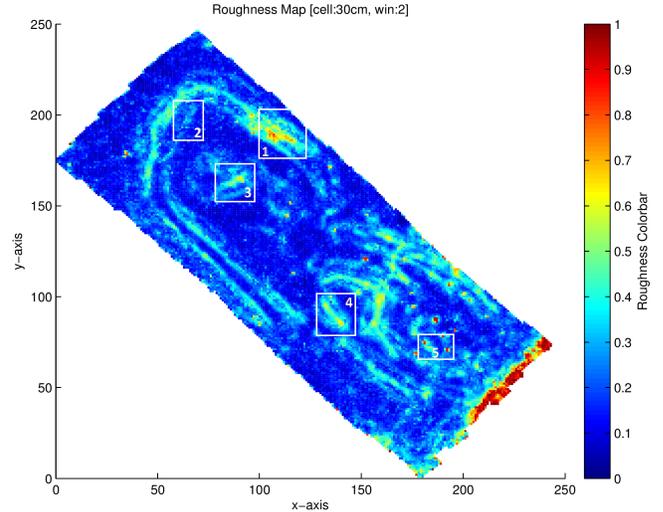


Fig. 7: Roughness Map of Marscape DEM

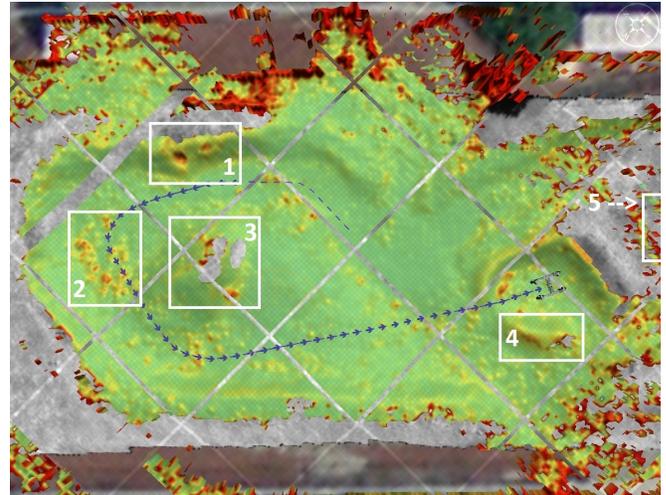


Fig. 8: Marscape Overhead

Marscape terrain is permanent except for the rocks, which are moved periodically. For this reason the rock features did not match the position exactly but the roughnesses were captured in both the DEM and the KRex LiDAR system.

C. Method Comparison

We compare our roughness map method with PTA. PTA uses an obstacle height threshold of $\delta = 0.3$ within its immediate neighbors. We use visual inspection to compare the methods since each method uses their own metric to define obstacles and roughnesses. The left hand column of Fig. 9 uses JSC's rockyard test site, and the right hand column uses JSC's crater test site to compare methods. The color scheme of PTA is based on the probability the terrain is an obstacle or not given by red and green— (obstacle and non-obstacle). The roughness map follows the color gradient of [red→ green] which is equivalent to [rough → smooth].

PTA is a binary obstacle detection algorithm that only detects changes in height as seen in Fig. 9a and Fig. 9b.

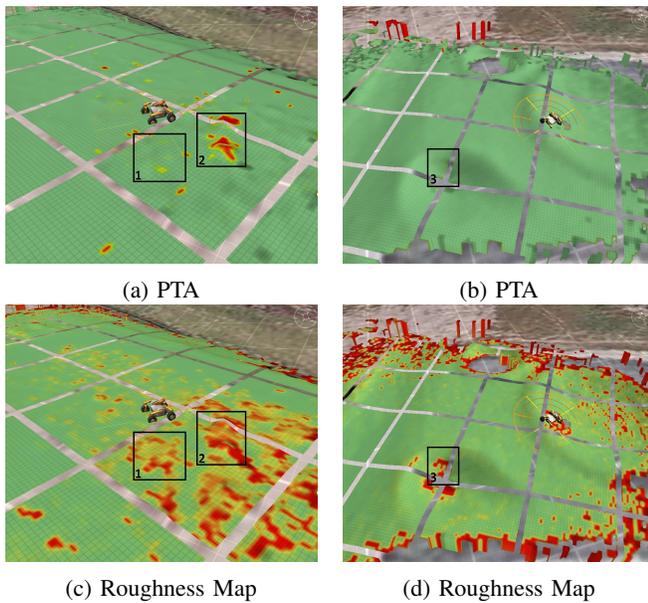


Fig. 9: Method Comparison PTA vs Roughness Map using JSC Rockyard (left column) and Crater (right column) data set

The roughness map detects the slightest changes in surfaces (Fig. 9c and 9d) thus providing more estimate of roughness. Another advantage of this algorithm is its resilience to terrain changes over time. The differences in methods are seen in boxes 1-3 of Fig. 9. PTA only detects several large rocks in box 2 of the JSC rockyard, whereas the roughness map detects all the rocks that are considered obstacles and rough. Within the crater data set, PTA detects no obstacles while the roughness map detected box 3 as steep; additionally the roughness map detected very small changes on the surface as shown by the scattered yellow blotches in Fig. 9d.

VI. CONCLUSION

We defined a new metric that measures terrain roughness for rover navigation. The roughness map looks at the terrain surface and is able to quantify roughness through the use of surface normals, projected plane heights, slopes, residuals, and coverage. This approach is able to detect and provide more information about the terrain than current existing methods. The effectiveness of the roughness map method was shown in the experimental results when compared to other methods.

One of the benefits of using surface normals is the robustness against time dependent plane height errors induced by inherent pose errors. This method is also adaptable to any four-wheeled systems, only requiring the system to know the wheel size and its roll/pitch threshold to adjust obstacle definitions ($\delta, \theta/\phi$) and window sizing (K). This method has a running time that takes at worst $O(2[(2K + 1)^2 - 1]N)$, which is linear time.

ACKNOWLEDGMENT

The authors would like to thank the Intelligent Robotics Group (IRG) at NASA Ames Research Center, especially Liam Pedersen who provided us with data, access to the KRex rover, access to and guided us through the inevitable questions of what would work in terms of exo-planetary rovers.

REFERENCES

- [1] M. Bajracharya, M. W. Maimone, and D. Helmick, "Autonomy for mars rovers: Past, present, and future," *Computer*, vol. 41, pp. 44–50, 2008.
- [2] J. Carsten, A. Rankin, D. Ferguson, and A. T. Stentz, "Global path planning on-board the mars exploration rovers," in *IEEE Aerospace Conference*, 2007.
- [3] C. Leger, A. Trebi-Ollennu, J. Wright, S. Maxwell, R. Bonitz, J. Biesiadecki, F. Hartman, B. Cooper, E. Baumgartner, and M. Maimone, "Mars exploration rover surface operations: Driving spirit at gusev crater," in *In Proceedings of the 2005 IEEE Conference on Systems, Man, and Cybernetics*, October 2005.
- [4] D. Stavens, G. Hoffmann, and S. Thrun, "Online speed adaptation using supervised learning for high-speed, off-road autonomous driving," *International Joint Conference on Artificial Intelligence*, 2007.
- [5] L. B. Cremean, T. Foote, J. H. Gillula, G. H. Hines, D. Kogan, K. L. Kriechbaum, J. C. Lamb, J. Leibs, L. Lindzey, C. E. Rasmussen, A. D. Stewart, J. W. Burdick, and R. M. Murray, "Alice: An information-rich autonomous vehicle for high-speed desert navigation: Field reports," *J. Robot. Syst.*, vol. 23, no. 9, June 2006.
- [6] M. Spenko, Y. Kuroda, S. Dubowsky, and K. Iagnemma, "Hazard avoidance for high-speed mobile robots in rough terrain," *J. Field Robotics*, vol. 23, no. 5, pp. 311–331, 2006.
- [7] S. Thrun, M. Montemerlo, and A. Aron, "Probabilistic terrain analysis for high speed," *Robotics Science and Systems II*, August 2006.
- [8] K. Iagnemma and S. Dubowsky, *Mobile Robots in Rough Terrain*, 2004.
- [9] A. El-Kabbany and A. Ramirez-Serrano, "Terrain roughness assessment for human assisted ugv navigation within heterogeneous terrains," *International Conference of Robotics and Biomimetics*, 2009.
- [10] S. Goldberg, M. Maimone, and L. Mattheis, "Stereo vision and rover navigation software for planetary exploration," *IEEE Aerospace Conference Proceedings*, March 2002.
- [11] M. Maimone, Y. Cheng, and L. Mattheis, "Two years of visual odometry on the mars exploration rovers: Field reports," *J. Field Robot.*, vol. 24, no. 3, pp. 169–186, Mar. 2007. [Online]. Available: <http://dx.doi.org/10.1002/rob.v24:3>
- [12] L. Pedersen, M. Allan, H. Utz, M. Deans, X. Bouyssounouse, Y. Choi, L. Flückiger, S. Y. Lee, V. To, J. Loh, W. Bluethmann, R. R. Burrige, J. Graf, and K. Hambüchen, "Tele-operated lunar rover navigation using lidar," 2012.