HORIZON BASED ORIENTATION ESTIMATION FOR PLANETARY SURFACE NAVIGATION

X. Bouyssounouse, A. V. Nefian, A. Thomas, L. Edwards, M. Deans and T. Fong

NASA Ames Research Center, SGT, Stanford University

ABSTRACT

Planetary rovers navigate in extreme environments for which a Global Positioning System (GPS) is unavailable, maps are restricted to relatively low resolution provided by orbital imagery, and compass information is often lacking due to weak or not existent magnetic fields. However, an accurate rover localization is particularly important to achieve the mission success by reaching the science targets, avoiding negative obstacles visible only in orbital maps, and maintaining good communication connections with ground. This paper describes a horizon solution for precise rover orientation estimation. The detected horizon in imagery provided by the on board navigation cameras is matched with the horizon rendered over the existing terrain model. The set of rotation parameters (roll, pitch yaw) that minimize the cost function between the two horizon curves corresponds to the rover estimated pose.

Index Terms— autonomous navigation, localization, horizon matching

1. INTRODUCTION

Localization of the rover and mapping of the surrounding terrain with high precision is critical to surface operations in planetary rover missions, particularly within subtasks such as rover traverse planning, hazard avoidance, and target approaching. Typically, planetary rover localization on past and current NASA missions relies on tedious manual matching of rover camera views and orbital maps [1]. Alternatively, the location of rovers can occasionally be verified by spacecraft imagery [2].

The system presented in this paper determines automatically the rover orientation by matching the horizon curve detected in the local imagery and matching it with the rendered horizon from a given rover pose in prior 3D terrain maps. Horizon detection methods for orientation estimation have been developed in the past for unmanned aerial vehicle (UAV) [3], [4], [5], micro aerial vehicle (MAV) [6] and marine applications [7]. The most successful approaches for horizon detection are based on edge detection and selection, image segmentation (clustering), image classification or a combination of the above methods. Specific to the planetary surface navigation are the unknown terrain features as seen from the rover/lander camera orientation and resolution. This makes classification based methods ([3], [7], [8], [9], [10]) difficult to use. A direct color based sky segmentation is also unreliable for images captured in various missions. It is known for example that the sky color varies strongly between Earth, Moon and Mars. In addition, in the presence of atmosphere, the horizon line can be fuzzy and affected by haze or dust. These conditions can confuse a direct edge detection and selection-based technique ([11], [12], [13], [14], [15]).

The horizon detection method presented in this paper is based on a Bayesian model for the image formation related to the work described in [16]. The model parameters are learned iteratively from a set of apriori statistics. The image formation model unifies under the same formulation both edge and texture-based approaches and reduces dependencies on apriori knowledge of the terrain and atmospheric conditions.

The statistical solution for horizon detection is described in Section 2. Section 3 describes an efficient multi-scale horizon rendering solution that is used to determine the optimal rover pose (Section 4). Finally the experimental results and conclusions of this work are described in Sections 5 and Section 6 respectively.

2. HORIZON DETECTION

The horizon detection method described in this paper uses low resolution monochrome images [17] to simulate navigation camera imagery in a mission analogue scenario. For the particular set of images discussed in this paper, each pixel corresponds to either a terrain region or sky region. Pixels that show the rover body can be easily masked given the known fixed camera pose with respect to the rover body. Furthermore there is a given structure to these images which can be exploited to improve the horizon detection. Assuming an upright camera orientation, a top to bottom scan of each image column will have at most one transition from sky to terrain regions. The transition pixel, if it exists, corresponds to the horizon pixel in each column. In practice the transition within a column from sky to terrain region can happen at any pixel with a different likelihood as it will be discussed later in this section. Furthermore, in natural environments horizon pixels in consecutive columns have small changes in row indices corresponding to a smooth horizon curve. The horizon detection problem is stated as finding the optimal global path of horizon pixels for each image column that satisfies the horizon smoothness constraint. The image formation model described above is represented as a Bayesian model in Figure 1. Rectangles represent discrete nodes and circles denote continuous nodes. Shaded rectangles represent hidden nodes while clear circles represent observations. Each pixel at location (col, row), in

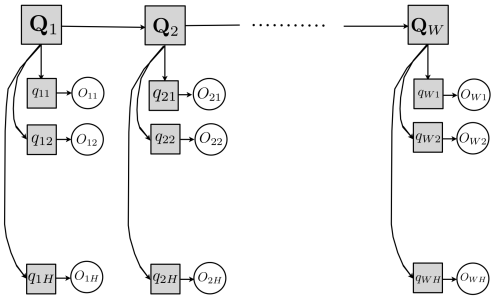

Fig. 1: Bayesian image formation model.

the image with H rows and W columns, is associated with an observation $O_{col,row}$. The hidden parent node $q_{col,row}$ for each observation has a binary value associated with either a "sky" or "ground" region. The discrete hidden node $\mathbf{Q}_{col}$ is the parent node for the set of nodes $\{q_{col,row} : row = 1 \dots H\}$, and its value is the column col where the transition from sky to ground region occurs.

The observation at each pixel is a two dimensional vector consisting of the image intensity at the specific pixel and the edge density. The edge density value is computed as the ratio between the number of Sobel edges in a window centered around the pixel over the total number of pixels in the given window.

The iterative horizon detection algorithm follows a set of steps described in Figure 2. Each box in this flowchart will be explained

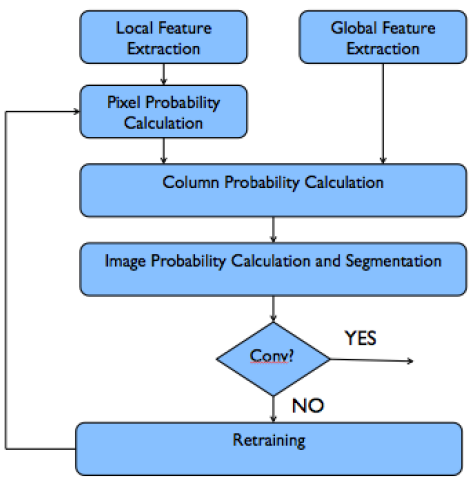

Fig. 2: Overview of horizon detection methods.

in more detail in the following steps.

Step 1. Pixel Probability Calculation.
The iterative method presented here starts with computation of observation probabilities at each pixel given the sky (s) or ground (g) regions $P(O_{ij}|q_{ij} = s)$ and $P(O_{ij}|q_{ij} = g)$. The method is initialized with a set of statistics associated with the sky (clear sky and cloudy subregions) and ground (well lit and shadowed subregions) regions.

At consecutive iterations the segmented image regions are used to re-learn the regions statistics using a random forest classifier as shown in Figure 2.

Next steps of the algorithm will consider dependencies among neighboring pixels.

Step 2. Image Column Probability Calculation.
The probability of a pixel in column j and row k being a horizon pixel is computed as:

$$P(\mathbf{O}_j|\mathbf{Q}_j = k) = P(O_{jk}|edge) \times \prod_{i=1}^{k} P(O_{ji}|q_{ji} = s) \prod_{i=k+1}^{H} P(O_{ji}|q_{ji} = g). \quad (1)$$

where $\mathbf{O}_j = O_{1j} \dots O_{Hj}$. $P(O_{ji}|q_{ji} = s)$ and $P(O_{ji}|q_{ji})$ were calculated in Step 1 and $P(O_{kj}|edge)$ is the likelihood that the pixel belongs to an image edge.

Step 3. Image Probability Calculation and Segmentation.
The optimal path $\underline{\mathbf{Q}} = \mathbf{Q}_1 \dots \mathbf{Q}_W$ that maximizes

$$\begin{aligned} P(\underline{\mathbf{O}}|\underline{\mathbf{Q}}) &= P(\mathbf{O}_1 \dots \mathbf{O}_W|\mathbf{Q}_1 \dots \mathbf{Q}_W) \\ &= \prod_{j}^{W} P(\mathbf{O}_j|\mathbf{Q}_j) \prod_{j=1}^{W} P(\mathbf{Q}_j|\mathbf{Q}_{j-1}), \end{aligned} \quad (2)$$

is computed via the Viterbi algorithm [18] and gives the optimal sequence of horizon pixels that separate the sky and ground regions. In Equation 2, $\underline{\mathbf{O}} = \mathbf{O}_1 \dots \mathbf{O}_W$ and $P(\mathbf{O}_j|\mathbf{Q}_j)$ are the probabilities calculated in Equation 1. The transition probability between horizon pixel locations in consecutive columns filters out variations in the horizon curve due to noise and it is calculated as:

$$P(\mathbf{Q}_j = k|\mathbf{Q}_{j-1} = l) = \begin{cases} \frac{1}{N} & \text{if } |k-l| \leq N \\ 0 & \text{otherwise} \end{cases}.$$

The value N is set based on assumptions to the horizon smoothness. For example a large value for N allows more abrupt changes in the horizon curve than a lower value for N. With the above formulation, finding the horizon curve is equivalent to finding the best sequence of $\mathbf{Q}_j$ in Equation 2. The efficiency of the optimal path computation is increased by considering only a subset of $P(\mathbf{O}_j|\mathbf{Q}_j)$ corresponding to highest likelihood values. The likelihood of all observations given the image formation model is approximated by the following equation

$$P(\underline{\mathbf{O}}) \approx \max_{\underline{\mathbf{Q}}} P(\underline{\mathbf{O}}, \underline{\mathbf{Q}}) = P(\underline{\mathbf{O}}|\underline{\mathbf{Q}})P(\underline{\mathbf{Q}}) \propto P(\underline{\mathbf{O}}|\underline{\mathbf{Q}}). \quad (3)$$

If the change in likelihood at consecutive iterations falls below a fixed threshold the iteration stop, and the final horizon curve is obtained (Figure 2). Otherwise, the model parameters are retrained as described in the next section.

Step 4. Iterative learning of region density function.
The observation vectors associated to ground and sky regions at each iteration are used to train a random forest classifier. The output of the classifier is used to compute the observation probabilities in the step 1 of the next iteration. In typical planetary missions the speed of the rover is of the order of 5-10 cm/s and the image statistics at consecutive frames don't change significantly. Under these conditions, and in order to reduce the onboard computational complexity the model trained on observations from previous images can be used to initialize the iterative procedure for the current frame In our experiments we use the most recent 50,000 observations collected from past images.

3. HORIZON RENDERING

Horizon rendering starts with the generation of a synthetic view at any given pose on the surface. This rendered view is obtained from the orbital digital elevation model (DEM) and the rover mounted camera intrinsic and extrinsic parameters. Finding the rendered horizon is equivalent to finding the curve that segments the ground and non-ground regions in the rendered image. The rendered horizon is determined by topographical features ranging from the immediate vicinity of the rover to regions that can extend tens of kilometers away from the camera. An accurately rendered horizon curve requires both high resolution and high coverage DEMs. This requirement leads memory limitations which are particularly important when the processing is offloaded to a Graphical Computational Unit (GPU) for fast rendering. To accommodate these constraints,

the terrain model is split spatially into multiple tiles at varying resolutions. The synthesized image is then computed from a selection of these tiles in the view frustum at resolutions based on camera to tile distances which guarantee a minimum projected image pixel resolution. In our experiments we used a low-coverage (.8×.8 km), high-resolution (1 m per post) terrain model, augmented with a large-coverage (10×10 km) low-resolution terrain (9 m per post). A multi-resolution representation with a resampling factor of two was chosen to capture both high resolution details and extended coverage. Figure 3 illustrates the hillshaded terrain at a test site that combines the low resolution high coverage with low coverage high resolution (inside the rectangle) terrain models. This approach satisfies both

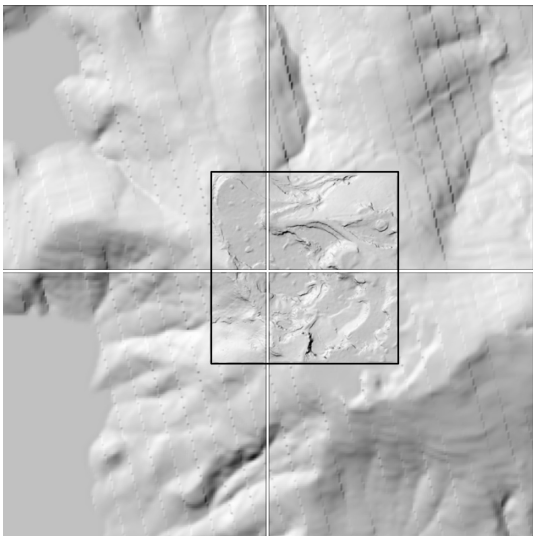

Fig. 3: Hillshaded terrain tiles at the field test site combining low resolution high coverage and high resolution low coverage area (inside the rectangle).

the wide coverage and high resolution requirements for horizon rendering, while accommodating the memory constraints of a typical GPU. The rendered image is computed using standard OpenGL libraries, [19] and the horizon curve is computed as the boundary of the rendered surface. Figure 4 illustrates a rendered image as seen by camera on board the rover for a given rover pose. The red area denotes the surface rendered by using the large coverage, low resolution orbital terrain model. This area would not be visible if only the low-coverage, high-resolution terrain were used, leading to an incorrect horizon curve.

4. HORIZON MATCHING

Let the detected horizon be $\mathbf{d}$ and the rendered horizon for given rover orientation ($\theta = \{roll, pitch, yaw\}$) be $\mathbf{r}(\theta)$. Both $\mathbf{d}$ and $\mathbf{r}(\theta)$ are vectors of size equal to image width W. The elements of these vectors contain the row index of the horizon curve for each column in the image. In horizon matching the goal is to determine the optimal rover orientation $\tilde{\theta}$ such that

$$\tilde{\theta} = \arg\min_{\theta} \sum_{i} (\mathbf{d}_i - \mathbf{r}_i(\theta))^2 \tag{4}$$

In case the ith rendered or observed column vector contains only sky or only ground, then this vector is considered invalid, and is omitted from the summation. The sum of squared differences is scaled by

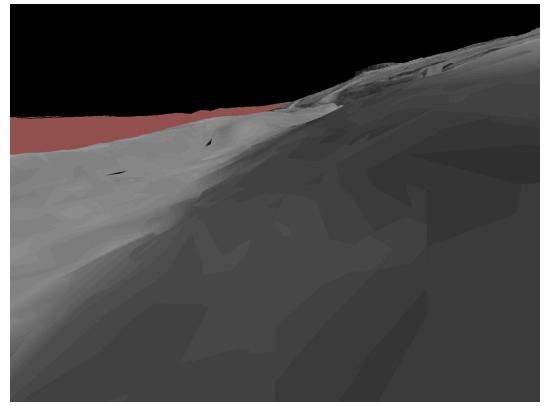

Fig. 4: Example of a rendered horizon image depicting both local (gray) and wide-area (red) models.

the ratio of horizontal image points W to number of valid points. In case the number of valid points drops below a threshold for an image then that image is not considered for comparison. Equation 4 is solved using Gauss Newton algorithm with numerical derivatives.

5. EXPERIMENTAL RESULTS

Data used to validate the described algorithms were gathered from a 3 km traverse of a rover operating in the Basalt Hills near San Luis reservoir, California at an average speed of 0.8 m/s. Figure 5 illustrates typical results for horizon detection (blue), horizon rendered from the INS unit (black), IMU and wheel odometry (red) and horizon matching (green). Note that the estimated rover pose generates a rendered horizon line that is closer to the actual horizon line than the horizon rendered from the INS or IMU pose. The spikes in the black and red curves are due to unfiltered noise in the orbital DEM. We ran two experiments to evaluate the method described in this paper. First, we measured the average pixel offset between the detected horizon line and the horizon line rendered for best rover orientation obtained using solution in Section 4. This experiment validates the matching technique. Figure 6 shows the average pixel offset for the detected vs INS horizon line and compares this to the average pixel offset for the detected vs horizon matched line, at each frame in the trajectory.

In the second experiment we estimated the orientation accuracy of the method including the horizon detection and matching techniques. We used as ground truth the INS rendered horizon line and assumed a set of noisy INS values as input for the horizon matching method. However, note that the INS orientation is often affected by noise to due rover movement or device inaccuracies and visual inspection showed that our method improves almost always INS orientation.

In order to reduce the effect of INS inaccuracy we used an orientation input noise uniformly distributed over ± 5 degrees in roll, pitch and yaw. At this noise level we found that the algorithm would occasionally converge to the wrong local minimum, and this issue was mostly mitigated by using multiple restart conditions of -2.5, 0.0, +2.5 degree offsets in roll, pitch, yaw, and choosing the lowest cost horizon line match from the resulting 27 combinations. Figure 7 illustrates the recovered pitch value (green) vs the input pitch value (red).

Nearly all of the large horizon matched errors in roll, pitch, and yaw were found to be due to the following conditions: high contrast clouds which were mistaken for part of the horizon; a camera position with a poor view of the horizon; large nearby rocks or dirt

mounds not sufficiently large to be captured in the DEM, yet close enough to block a portion of the horizon.

Fig. 5: Detected horizon (blue), rendered horizon from INS (black), IMU and wheel odometry (red) poses. The green curve shows the rendered horizon using the pose estimation method described in this paper.

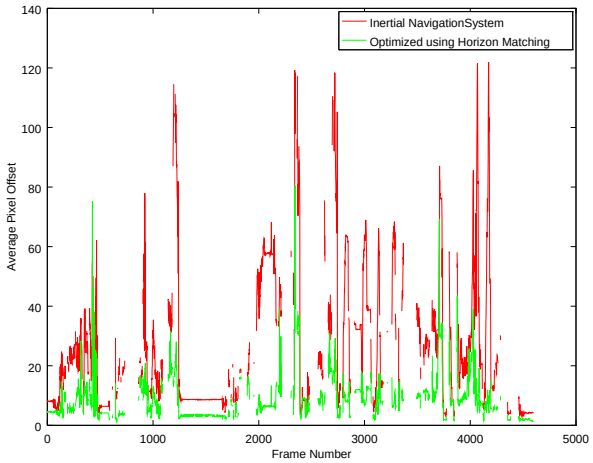

Fig. 6: Average pixel offset magnitude for detected vs INS (red), and detected vs horizon matching (green).

6. CONCLUSIONS

This paper describes a system for planetary rover orientation estimation by matching the horizon detected in images captured by the rover navigation camera and the horizon rendered from an existing terrain model obtained from satellite imagery. The orientation estimate is obtained using Gauss Newton method that minimizes the least square distance between the vectors describing the detected and rendered horizon lines. The horizon rendering solution uses an efficient multi-resolution large scale terrain map obtained from orbital imagery. The horizon detection method uses a Bayesian image formation model and an iterative statistical algorithm to estimate the optimal segmentation curve between terrain and sky regions. The results have been successfully tested in multiple environments and illumination conditions including terrestrial analogues sites.

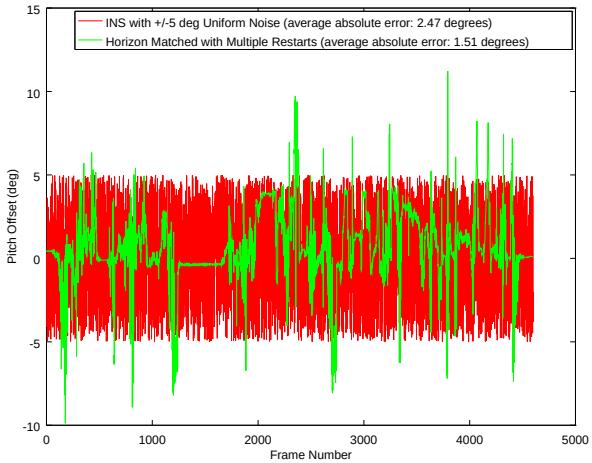

Fig. 7: INS with $\pm 5^\circ$ uniform random noise added (red), and resulting horizon optimized roll errors (green).

7. REFERENCES

[1] T. J. Parker, M. C. Malin, F. J. Calef, R. G. Deen, H. E. Gengl, M. P. Golombek, O. Pariser J. R. Hall and, M. Powell, R. S. Sletten, and the MSL Science Team., "Localization and contextualization of curiosity in gale crater, and other landed mars missions," *Lunar and Planet. Sci. Conf. 44th*, 2013.

[2] Rongxing Li, Shaojun He, Yunhang Chen, Min Tang, Pingbo Tang, Kaichang Di, Larry Matthies, Raymond E Arvidson, Steven W Squyres, Larry S Crumpler, et al., "MER Spirit rover localization: Comparison of ground image–and orbital image–based methods and science applications," *Journal of Geophysical Research: Planets (1991–2012)*, vol. 116, no. E7, 2011.

[3] Nasim Sepehri Boroujeni, S Ali Etemad, and Anthony Whitehead, "Robust horizon detection using segmentation for uav applications," in *Computer and Robot Vision (CRV), 2012 Ninth Conference on*. IEEE, 2012, pp. 346–352.

[4] Timothy G McGee, Raja Sengupta, and Karl Hedrick, "Obstacle detection for small autonomous aircraft using sky segmentation," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 4679–4684.

[5] Saul Thurrowgood, Dean Soccol, Richard JD Moore, Daniel Bland, and Mandyam V Srinivasan, "A vision based system for attitude estimation of uavs," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 5725–5730.

[6] Scott M. Ettinger, Michael C. Nechyba, Peter G. Ifju, and Martin Waszak, "Towards flight autonomy: Vision-based horizon detection for micro air vehicles," 2002.

[7] Sergiy Fefilatyev, Volha Smarodzinava, Lawrence O Hall, and Dmitry B Goldgof, "Horizon detection using machine learning techniques," in *Machine Learning and Applications, 2006. ICMLA'06. 5th International Conference on*. IEEE, 2006, pp. 17–21.

[8] C. De Wagter G.C.H.E. de Croon, B.D.W. Remes and R. Ruijsink, "Sky segmentation approach to obstacle avoidance," *Aerospace Conference Proceedings, IEEE*, 2011.

[9] Ali Pour Yazdanpanah, Emma E Regentova, Ajay Kumar Mandava, Touqeer Ahmad, and George Bebis, "Sky segmentation by fusing clustering with neural networks," in *Advances in Visual Computing*, pp. 663–672. Springer, 2013.

[10] Cyrus Minwalla, Kyle Watters, Paul Thomas, Richard Hornsey, Kristopher Ellis, and Sion Jennings, "Horizon extraction in an optical collision avoidance sensor," *24th Canadian Conference on Electrical and Computer Engineering*, 2011.

[11] Damien Dusha, Wageeh Boles, and Rodney Walker, "Fixed-wing attitude estimation using computer vision based horizon detection," *12th Australian International Aerospace Congress*, 2007.

[12] Bahman Zafarifar and Hans Weda, "Horizon detection based on sky-color and edge features," *Visual Communications and Image Processing*, vol. 6822, 2008.

[13] YF Shen, D Krusienski, J Li, and Z Rahman, "A hierarchical horizon detection algorithm," *Geoscience and Remote Sensing Letters, IEEE*, 2013.

[14] Ting-Chih Lin Wen-Nung Lie, Tom C.I. Lin and Keng-Shen Hung, "A robust dynamic programming algorithm to extract skyline in images for navigation," *Pattern Recognition Letters*, 2005.

[15] Touqeer Ahmad, George Bebis, Emma Regentova, Ara Nefian, and Terry Fong, "Coupling dynamic programming with machine learning for horizon line detection," *International Journal on Artificial Intelligence Tools*, vol. 24, no. 4, pp. 1540018, 2015.

[16] Ara Nefian and Gary Bradski, "Detection of drivable corridors for off-road autonomous navigation," in *International Conference on Image Processing*, 2006.

[17] Allied Vision Technologies, "GigE vision camera with Sony ICX267 and PoE option," http://www.alliedvisiontec.com/emea/products/cameras/gigabit-ethernet/manta/g-146bc.html.

[18] G David Forney Jr, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[19] Dave Shreiner and The Khronos OpenGL ARB Working Group, *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 3.0 and 3.1*, Addison-Wesley Professional, 7th edition, 2009.