Student Name: *Zili Ge* Student ID#: *1003213845*

DepthMesh: Obstacle Detection using Surface Normals

Project Problem Definition

Mars is an unforgiving landscape, with many perils facing a robotic rover. A rover moving at an appreciable speed or autonomously from waypoint to waypoint cannot be monitored by an operator in real time, due to the limitation of the speed of light; the round-trip speed of light to Mars can be upwards of 40 minutes. As a result, autonomous rovers on the surface must be able to identify and detect hazards on the surface of the planet, lest risking damage from collisions. We propose an algorithm to perform obstacle segmentation using a pair of stereo cameras. This approach is not learning based, making failure modes easier to analyze and correct.

Due to the limited depth resolution available the original problem statement of path planning has been scaled down to obstacle detection, as the model has trouble identifying rocks more than 5-10 meters away from the camera. The problem statement of identifying flat but disconnected and therefore untraversable terrain is still preserved from the proposal.

The model begins by generating a disparity map using stereo cameras, which is used to construct a point cloud. Surface normals are estimated and a mesh is constructed, where connected-component labelling is utilized to identify terrain reachable by the rover given its current position.

Project Methodology

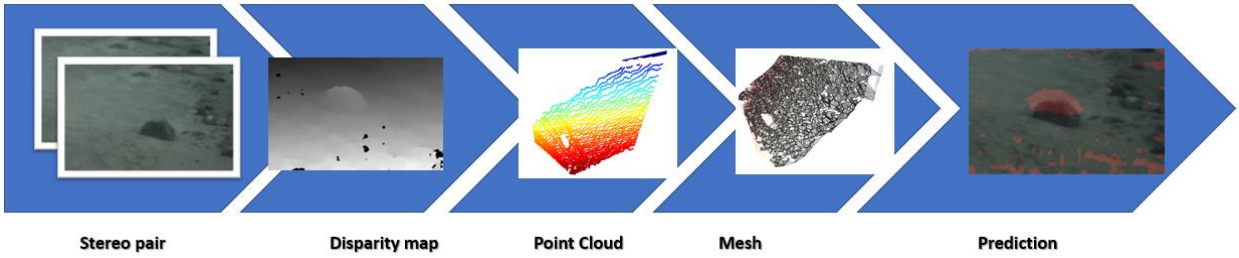

Figure 1: Proposed pipeline for generating disparity map.

The project pipeline begins with undistorting and rectifying a set of images from a stereo pair. A disparity map is then generated using semi-global matching [1]. Results are displayed in Figure 2; note that the method generates smooth, consistent disparity maps for nearby objects, successfully picking out the rock in the foreground as well as the rock wall to the right of the image. This method does not work well for distant objects and generates speckling artifacts in the sky. As a result, the disparity map is limited to the ground, which is currently determined by using only the bottom pixels of the image. Finally, a de-speckling algorithm is applied to the output of the depth map.

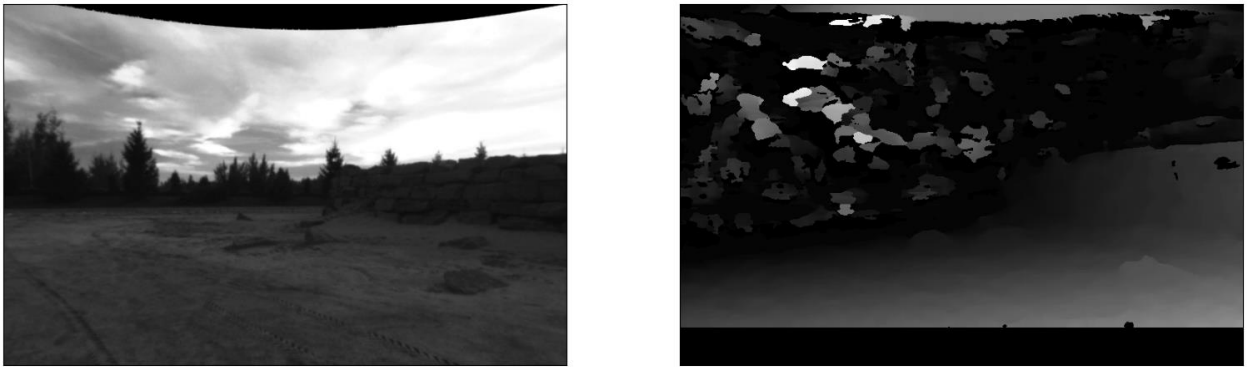

Figure 2: The rectified image vs the disparity map. Notice that the rock on the bottom left is clearly visible, as is the rock wall towards the center right of the image.

After generating the disparity map, we assume a pinhole model of the camera and use the baseline distance as well as the focal lengths in order to compute a point cloud from the disparity map. The point cloud is then uniformly downsampled to keep computational time for the following steps reasonable. After downsampling, surface normals for each point are estimated by applying principal component analysis (PCA) on the nearest 30 points to each sample point, with the surface normal pointing towards the smallest principal component [2]. Finally, a consistent global orientation for all surface normals are gathered by propagation of an initial seed orientation using a spanning tree [3].

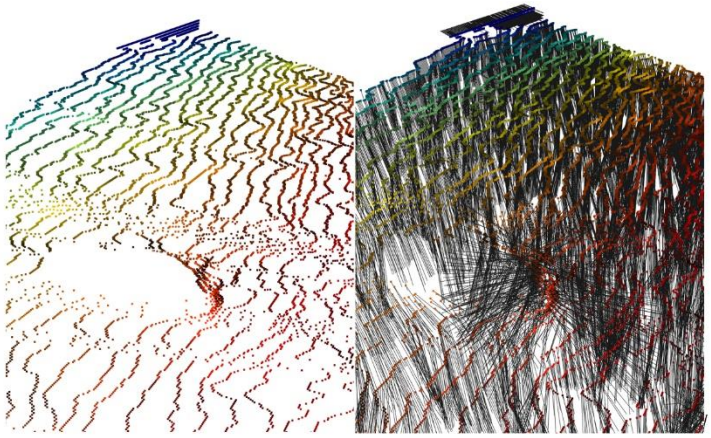

Figure 3: Point cloud on the left, with estimated surface normals on the right. Notice the surface normals belonging to an obstacle being wildly different from that belonging to the terrain.

Using point clouds and a surface normal, surface reconstruction is performed with the ball-pivoting algorithm [4], which scans each point in the point cloud with several different radii and constructs a triangle between three points if a ball of each radius contains exactly two other points. The resultant mesh is smoothed using Laplacian smoothing. This gives us a way to transform the point cloud to a graph structure by taking triangle edges as the edges of the graph and points as vertices. This allows us to perform connected component analysis [5]. To do this, we first identify surface normals belonging to an obstacle instead of the ground; this is done simply by removing points with normals that are too far away from the median normal, or what is assumed to be the ground plane. Finally, the connected components algorithm is used to identify the maximal connected subgraph of this mesh, which is assumed to be the current position of the robot. Note that this algorithm has the ability to identify flat but unreachable terrain as an obstacle, something that can be missed by a naïve solution. All vertices not a member of the maximal connected subgraph are removed and assumed to be part of an obstacle.

Remaining vertices are now reprojected back into the rectified image, with a maximum pixel intensity for the closest pixel in the original image. Gaussian blur is then applied to these pixels. Finally, we arrive at the obstacle mask by binarizing the blurred pixels.

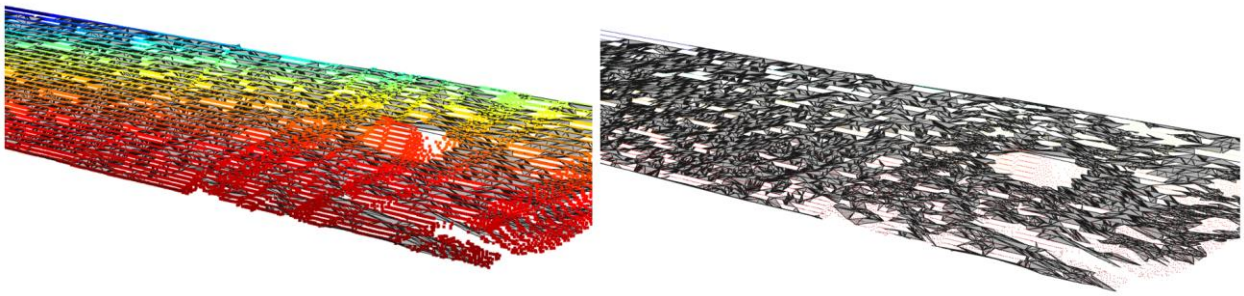

Figure 4: Generated mesh with overlaid point cloud on the left, followed by the mesh by itself on the right. Notice how the rock has its meshes removed.

This particular methodology was chosen out of a desire to have an independent object detection tool with a model that is fully explainable; modern machine learning approaches, although incredibly powerful, often lack explanation for misclassifications. This method uses a very geometrically motivated argument of thresholding surface normals, making it ideal as a tool for tasks that prioritize safety; crashing a rover on the surface of Mars would be costly! In addition, the formulation allows for several convenient tunable parameters, such as the maximum allowed slope. This can be changed simply by setting the threshold on the orientation of the surface normal away from tangent to the ground plane.

Libraries used were OpenCV *for image rectification, disparity map generation and despeckling of the disparity map, while* Open3D *was used for estimating surface normals, constructing the mesh, finding the maximal connected subgraph and Laplacian smoothing. For evaluation,* Scikit-Learn *was used to generate a confusion matrix and* Pandas *was used for writing results to a csv.* Matplotlib *and* Open3d *were used for data visualization.*

Project Evaluation and Results

Figure 5: The test images used. All images were taken from omni cam 0 of run 4. The frames from left to right are 900, 1080 and 1600.

To evaluate the validity of this obstacle detection framework, ground truth images for obstacles were labelled by hand. The labelled images were cropped to the detection region. Due to the difficulty in generating ground truth images, the test dataset is small and only consists of 3 images, although care was taken to make sure these test cases are representative and contain difficult sections. The test images are shown in Figure 5. The middle image has a barrier blocking a large part of the map, while the right image has a large stone wall that does much of the same function. All images have obstacles both in the foreground and background. The right image not only has very little traversable terrain available to the camera, it is also severely underexposed, which tests the disparity map generator.

Figure 6: An example of a ground truth mask (left) vs the original image (right).

In the dataset, the obstacle mask is drawn over rocks as well as terrain that is unreachable, at least when seen from the camera frame. Figure 6 demonstrates this obstacle mask; this image contains several boulders as well as a curb. The boulders are labelled as obstacles. Although the road is flat, there is no way for the rover to safely traverse onto that section of terrain due to the curb itself; as a result, it is labelled as an obstacle as well.

Qualitative Results

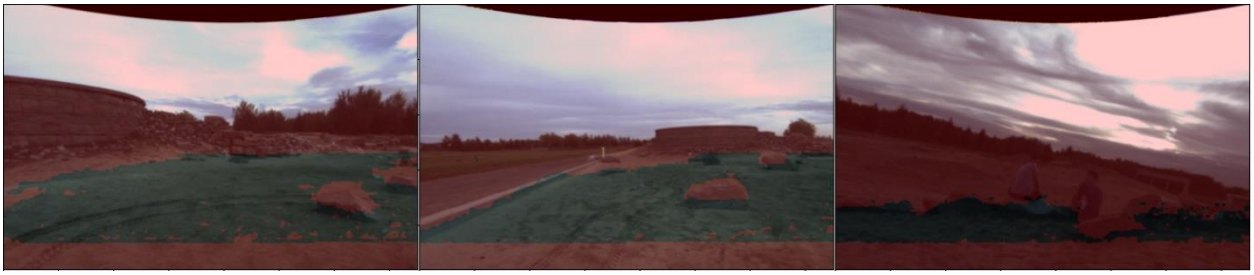

Figure 7: Generated masks. The masks are shown in red. Note the bottom red mask due to the disparity, as well as the top which is truncated.

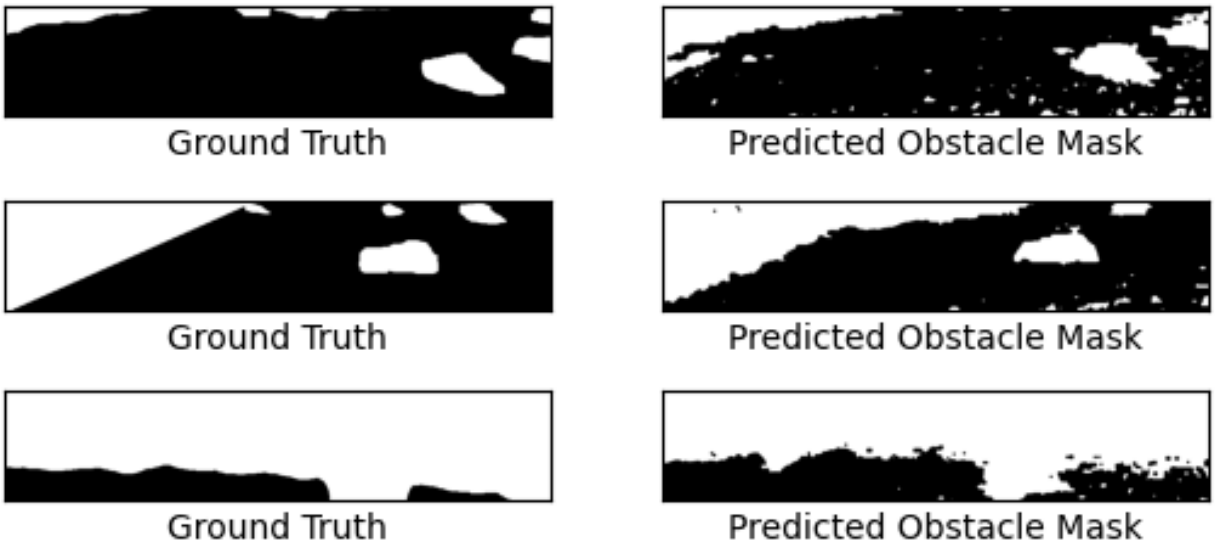

Figure 8: Binarized ground truth and predicted mask, focusing on the predictive region. The black represents the traversable region.

The model is able to generate convincing object masks for nearby obstacles, where all rocks in the foreground were correctly segmented. The rock walls in images 1 and 3 were correctly segmented, with the flat terrain separated by the curb in image 2 as well as the flat land beyond the rock wall in image 3 correctly identified as unreachable. However, faraway obstacles are not always detected; in the top image, for example, the third boulder in the top right of the middle image was not detected by the algorithm.

The algorithm also tends to mark the top left corner of the terrain as untraversable; this likely has to do with the camera itself, as this corner is the furthest away from the camera and the mesh generation algorithm has trouble generating meshes for these regions. This can be seen in the top and middle images, where the top left obstacle mask is significantly bigger than the ground truth.

Quantitative Results

Two metrics were used to determine the efficacy of this model. The first method, Intersection over Union (or IOU), is defined as the intersection of the predicted obstacle mask with the ground truth obstacle mask, over the union of the predicted and ground truth obstacle masks. We use IOU as it is a common and intuitive evaluation metric for segmentation models. A perfect segmentation has an IOU of 1, where the union of the segmentation is equal to its intersection, while the worst possible score is a 0, where there is no overlap between the predicted and actual masks. A visualization of the IOU is presented in figure 9.

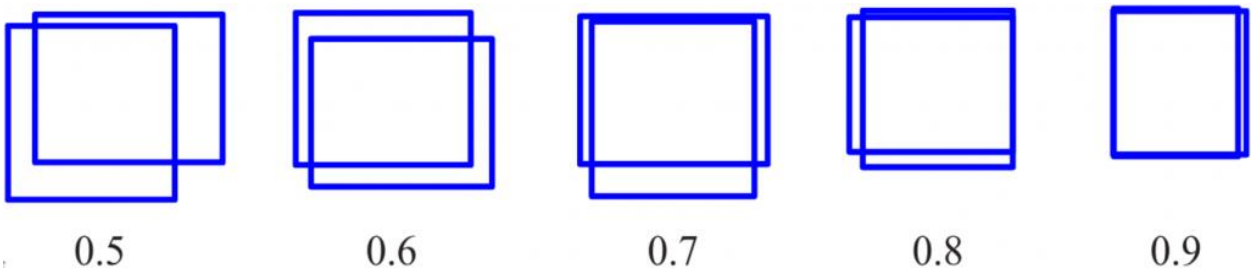

Figure 9: An illustration of goodness of fit for a given IOU value [6].

As this model is intended to be a safety mechanism, we wish to be as conservative as possible, minimizing the part of the obstacle mask misclassified as traversable terrain. As a result, an important metric is to determine the amount of misclassification of the obstacle mask as not an obstacle mask, or the sensitivity. We use $Sensitivity = \frac{TP}{TP+FN}$, where TP is the true positive pixels and FN is the false negative pixels. The best possible sensitivity is 1 where all pixels are successfully classified, where the worst is 0. The results are shown in table 1.

Table 1: Results for the test images.

Frame # (Omni Camera 0)	**IOU**	**Sensitivity**
900	0.317	0.376
1080	0.710	0.798
1600	0.828	0.997
Average	0.617	0.724

Surprisingly, frame 900, chosen as an easy frame, achieved by far the worst results in this dataset with the hardest image, frame 1600 achieving the best. This may have been an artifact of the dataset itself; as the percentage of obstacles is small, a wrongly segmented mask would result in a large error for both metrics. It appears that the two foreground rocks, accounting for most of the masked pixels had inaccurate masks which lowered the sensitivity. The remaining two images, with a large obstacle mask, fared better in this regard.

Challenges Faced

There were several challenges faced during this exercise.

- The disparity map pipeline took quite some effort to tune correct to get good results. Before settling on the omnidirectional cameras, attempts were made to use dense optical flow between successive images of the monocular camera, followed by triangulation using estimated pose and orientation from GPS coordinates and ground truth rover orientation to generate a depth map from the optical flow.
- There was a large difference between the distances represented by the depth map. The distance between two pixels near to the camera is much smaller than the distances between two pixels further away, resulting in failed mesh generation. Attempts were made at nonlinear distance functions when converting from a disparity map to a modified depth map, before finally settling on adding a constant value to the disparity, so taking its reciprocal results in a smaller range of values while still preserving linearity.

References:

[1] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328-341, 2008. Available: 10.1109/tpami.2007.1166.

[2] Z. Yu, T. Wang, T. Guo, H. Li and J. Dong, "Robust point cloud normal estimation via neighborhood reconstruction", *Advances in Mechanical Engineering*, vol. 11, no. 4, p. 168781401983604, 2019. Available: 10.1177/1687814019836043.

[3] S. König and S. Gumhold, "Consistent Propagation of Normal Orientations in Point Clouds", *Proceedings of the Vision, Modeling, and Visualization Workshop 2009*, 2009. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.441.5147&rep=rep1&type=pdf. [Accessed 21 December 2020].

[4] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva and G. Taubin, "The ball-pivoting algorithm for surface reconstruction", *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349-359, 1999. Available: 10.1109/2945.817351.

[5] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao and Y. Chao, "The connected-component labeling problem: A review of state-of-the-art algorithms", *Pattern Recognition*, vol. 70, pp. 25-43, 2017. Available: 10.1016/j.patcog.2017.04.018.

[6] T. Mai, "A Guide to State of the Art Object Detection for Multimodal NLP", *Natural Language Processing Blog*, 2020.