

Horizon Line Estimation In Glacial Environments Using Multiple Visual Cues

Stephen Williams and Ayanna M. Howard

Abstract—While the arctic possesses significant information of scientific value, surprisingly little work has focused on developing robotic systems to collect this data. For arctic robotic data collection to be a viable solution, a method for navigating in the arctic, and thus of assessing glacial terrain, must be developed. Segmenting the ground plane from the rest of the image is one common aspect of a visual hazard detection system. However, the properties of glacial images, namely low contrast, overcast sky, and cloud, mountain, and snow sharing common colors, pose difficulties for most visual algorithms. A horizon line detection scheme is presented which uses multiple visual cues to rank candidate horizon segments, then constructs a horizon line consistent with those cues. Weak cues serve to reinforce a selected path, while strong cues have the ability to redirect it. Further, the system infers the horizon location in areas that are visually ambiguous. The performance of the proposed system has been tested on multiple data sets collected on two different glaciers in Alaska, and compares favorably, both in terms of time and classification performance, to representative segmentation algorithms from several different classes.

I. INTRODUCTION

Though many scientists believe the condition of the giant ice sheets in Greenland and Antarctica are a key to understanding global climate change, there is still insufficient data to accurately predict the future behavior of those ice sheets. While satellites are able to map the ice sheet elevations with increasing accuracy, data about general weather conditions (i.e. wind speed, barometric pressure, etc.) must be measured at the surface. In order to obtain these measurements, human expeditions must be sent to these remote and dangerous areas. Alternatively, a group of autonomous robotic rovers could be deployed to these same locations, mitigating the cost, effort, and danger of human presence.

Given the arctic possesses significant information of scientific value, surprisingly little work has focused on developing robotic systems to collect this data. For arctic robotic data collection to be a viable solution, a method for navigating in the arctic, and thus of assessing glacial terrain, must be developed. Previous work has explored visual methods of assessing glacial hazards [1] and augmenting localization sensors [2] as part of a larger navigational system. Implicit in this work is the ability to segment the foreground terrain from distant mountains and the sky. Foreground segmentation

and horizon line estimation are essential aspects of many ground-based [3], [4] and aerial [5] robotic systems. Such segmentation allows processing to be focused on just the area of significance to the rover, reducing the overall computational requirements. Further, foreground segmentation acts as a first pass for obstacle detection; image regions which differ significantly from the ground plane are probably obstacles. Finally, dynamic elements, such as clouds, that could negatively impact visual odometry results can be removed from consideration, while the horizon itself can help in estimating the robot's orientation.

Common segmentation methods use information local to the examined pixel to make segmentation decisions. However, the properties of glacial images make local examination problematic. Overcast skies, common in glacial environments, often share the same color range as the ground plane snow. Further complicating segmentation, the clouds and ground plane often intersect visually, making the determination of the horizon difficult. When analyzing these images, humans tend to scan for visual cues in the form of strong horizon line segments. These line segments are then extended into image regions where the horizon determination is more ambiguous. This paper presents a computationally tractable horizon line extraction process which employs a similar strategy to infer the horizon line location in areas which exhibit weak visual cues. Section II briefly describes the types of glacial terrain encountered during field trials on two different glaciers in Alaska. Section III describes common approaches to ground segmentation, while Section IV details the proposed horizon line extraction algorithm. The horizon line results from the field tests are compared with other existing segmentation methods in Section V. Finally, conclusions and future work are discussed in Section VI.

II. GLACIAL ENVIRONMENTS

The main area of glaciers are largely flat and covered with snow, though the surface can be influenced by the underlying terrain and nearby mountains. Visible mountain peaks tend to be snow-covered as well, making the differentiation between the passable terrain and the vertical mountain sides difficult. Overcast skies, a common condition in glacial areas, share the same color range as the ground plane snow. Further complicating segmentation, the clouds and ground often intersect visually, making the localization of the horizon difficult. In extreme cases, known as "white out" events, the cloud-ground boundary becomes indistinguishable, making travel dangerous.

This work was supported by the National Aeronautics and Space Administration under the Earth Science and Technology Office, Applied Information Systems Technology Program

Stephen Williams and Ayanna M. Howard are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, 30332 swilliams8@gatech.edu, ayanna.howard@ece.gatech.edu

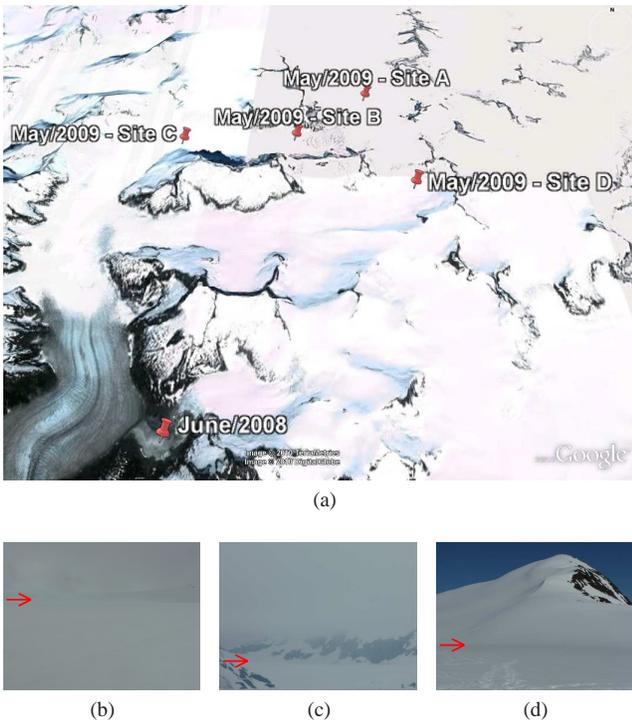


Fig. 1: (a) A map of the relative positions of the various test sites on Mendenhall Glacier during the June 2008 and May 2009 field tests. Test site on Lemon Creek Glacier is not shown. (b)-(d) Sample images from Lemon Creek Glacier and Mendenhall Glacier showing obscured horizons. For clarity, the horizon line is indicated on each sample image.

In June 2008 and May 2009, field trials were conducted on Mendenhall Glacier and Lemon Creek Glacier near Juneau, Alaska. Low cloud elevations limited travel to the glacier surface in 2008, but favorable weather conditions in 2009 allowed multiple tests sites to be utilized. Fig. 1 shows the location of each test site on the glacier, as well as images of the typical terrain.

At each test site, a set of salient still images were acquired, and video sequences were recorded from a camera on-board a custom, weather-hardened robotic platform. Individual maneuvers included small closed loops on flat terrain, long linear runs up significant terrain inclines, and switchbacks running down the side of mountain peaks. In total these recordings represent over 50,000 individual frames, and account for over a kilometer of traversal distance.

III. BACKGROUND

Little direct research exists in the area of glacier image processing. One project, sponsored by the Association of European Research Establishments in Aeronautics (EREA), uses vision to extract the foreground from glacial images as part of an automated snowcat convoy system [6]. It is assumed that the majority of the image is filled with the snowy region. Consequently, in the histogram of the image, the largest peak should be associated with the grayscale values of this region. An adaptive threshold based on the

boundaries of this peak is then used to separate the region of interest from unwanted objects and areas. This technique represents the only known prior art in the area of glacial foreground segmentation.

Generic image segmentation algorithms can also be applied to the foreground segmentation problem. The MUCAR project at the University of the Bundeswehr Munich uses heuristic criteria to separate roadway pixels from its surroundings [7]. Similar color-based segmentation methods have been used successfully as elements of robotic ground plane segmentation systems [3], [4], [8]. Others find the addition of texture information to be useful in the segmentation process [9], [10]. Region growing methods are common in segmentation algorithms, as they allow flexibility in the pixel test condition. Additionally, the output consists of connected regions, an advantage over thresholding and other per-pixel evaluation methods.

In the case of region growing, a merging criteria must be supplied which ultimately determines the system performance. Manually tuning these threshold values can be tedious. In contrast, the road segmentation method used by Stanley, the DARPA Grand Challenge entry from Stanford, uses a learned Gaussian mixture model to classify each image pixel by color [11], avoiding the need to set hard limits. Similarly, a water-sky segmentation system uses color and various texture measurements to successfully label each pixel, a problem visually similar to the glacial segmentation problem under consideration [12].

Fig. 2 illustrates a selection of these methods applied to a sample glacial image. The adaptive histogram threshold method successfully includes the ground region, but is unable to differentiate it from the sky. This leads to excessive error in the segmentation. The region growing method using color and texture information achieves slightly better performance, since the output consists of a single connected region. However, a single classification error at the horizon boundary can allow the output region to grow into the sky. Finally, the learning method, which uses a variety of image properties, such as smoothness, uniformity, and entropy, to classify each pixel using a Gaussian mixture model, still produces an unacceptably large classification error.

IV. HORIZON LINE EXTRACTION

All of the previous methods mentioned use information local to the examined pixel to make segmentation decisions. However, the properties of glacial images make local examination problematic. Overcast skies, common in glacial environments, often share the same color range as the ground plane snow. Fig. 3 shows an example of this phenomenon in which a section of the ground-cloud boundary has been magnified. Using only the information within the magnified boxed, it is difficult, if not impossible, to find the true horizon line. When analyzing these images, humans tend to look for strong horizon line segments somewhere in the image. These segments are then assumed to extend into the more ambiguous regions. Using this type of strategy, a ground segmentation method has been devised.

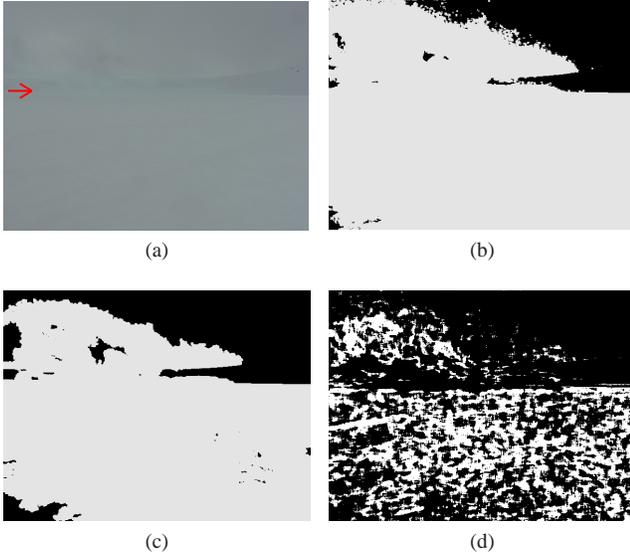


Fig. 2: (a) A sample image of Lemon Creek Glacier during overcast weather with the horizon line indicated by a red arrow. (b)-(d) The classification results of adaptive histogram thresholding, region growing, and Gaussian mixture model algorithms respectively. Each method produces a large classification error due to the similar color and texture properties of the snow and overcast sky.

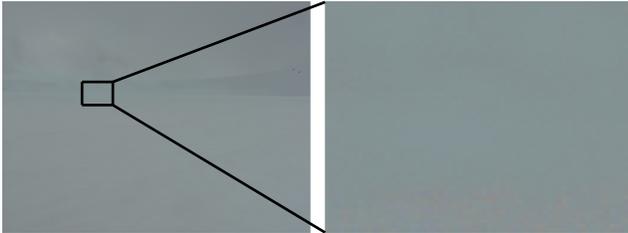


Fig. 3: A sample image of Lemon Creek Glacier during overcast weather with an enlarged region including a section of the horizon. The horizon line can be seen in the whole image, but nearly invisible in the enlarged section.

First, a set of nonparametric rank-order statistics are generated within a seed region, S . By constructing a pixel intensity histogram of this region, the median intensity, quartile values, and center 95% range can be efficiently calculated. The region S itself is formed by a static trapezoid in the bottom third of the image. This region, which is directly in front of the rover, is considered likely to contain ground plane data. This method is similar to the obstacle detection system presented in [13].

Strong line segments are then extracted from the image. A Canny edge detector [14] is used to find dominate image edges. The upper Canny threshold value is selected dynamically to classify 20% of image pixels as edges, with the lower threshold set to a fixed 40% of the upper threshold value. Each edge is then simplified to be piecewise linear using the Ramer-Douglas-Peucker algorithm [15]. A minimum

segment length constraint is enforced to remove the large number of noise-induced edges. A set of heuristic properties are then calculated for each remaining candidate segment, designed to test the likelihood that the candidate is actually part of the true horizon line. These properties are summarized in the following.

Segment Length

Longer line segments are more likely to be part of larger structures, such as the horizon or mountain boundaries, and less likely to come from localized surface texture. Eq. (1) weights longer segments more heavily.

$$W_{len} = Length_{segment}/Width_{image} \quad (1)$$

Color Below Segment

If a segment is part of the horizon line, then the region below the line segment should be statistically similar to the foreground seed region. A pixel intensity histogram is constructed from an area, B , immediately below each candidate. Using this histogram, the quartile intensity values are calculated and compared to the seed region statistics. The normalized euclidean distance between the quartile values of the seed region and the quartile values of the area below the line segment is used to weight each segment in (2).

$$W_{below} = 1 - \alpha \cdot \|Q_B - Q_S\| \quad (2)$$

where Q_i represents the vector of quartile boundary intensities, α is a normalization constant set to the inverse of the maximum quartile boundary difference, S is the seed region, and B is a small area below the current candidate segment.

Color Above Segment

If the segment is part of the horizon line, then the area above the line should be statistically different from the foreground seed region. In a similar fashion to the ‘Color Below Segment’ property, the euclidean distance between the quartile values of the seed region and the quartile values of the region, U , above the line segment is calculated. The property weight is given by (3).

$$W_{above} = \beta \cdot \|Q_U - Q_S\| \quad (3)$$

where Q_i represents the vector of quartile boundary intensities, β is a normalization constant set to the inverse of the maximum quartile boundary difference, S is the seed region, and U is a small area above the current candidate segment.

Color Column

Line segments are often generated at the upper edge of snow-covered mountain peaks, or at cloud-sky boundaries. This results in snow-colored pixels directly below a line segment, even though a section of non-white pixels exists between the line segment and the ground. By weighting each line segment by the percent of white pixels between the segment and the bottom of the image, these types of segments may be removed. The exact weight calculation method is given in (5).

$$A(i, j) = \begin{cases} 1 & \text{if } Q_S[1] < I(i, j) < Q_S[3] \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$W_{column} = \frac{1}{L} \sum_{i=x_1}^{x_2} \sum_{j=y_{seg}(i)}^{Height_{image}} A(i, j) \quad (5)$$

where L is the total number of pixels below the candidate segment, (x_1, y_1) and (x_2, y_2) are the end points of the current candidate segment, the function $y_{seg}(x)$ returns the y -value of the candidate segment at x , $Q_S[1]$ and $Q_S[3]$ are the first and third quartile boundary intensities of the seed region, respectively, and $I(i, j)$ is the image intensity.

Distance From Predicted Horizon

The position of the horizon location can be estimated using the current pose estimate of the camera, assuming a flat ground plane. The distance from this estimate to a candidate segment is used as a measure of the likelihood the candidate is part of the horizon. A Gaussian kernel centered on the horizon estimate is used as the weight, shown in (6). An appropriate variance for the Gaussian kernel can be estimated by multiplying the horizon position uncertainty caused by the uncertainty of the robot pose, with the horizon position uncertainty caused by the terrain variation from the assumed ground plane. If no prior knowledge of the terrain is available, a wide Gaussian should be used.

$$W_{dist} = \frac{1}{L} \sum_{i=x_1}^{x_2} G_{\sigma}(y_{est}(i) - y_{seg}(i)) \quad (6)$$

where the function $y_{seg}(x)$ returns the y -value of the candidate segment at x , $y_{est}(x)$ returns the y -value of the horizon estimate at x , $G_{\sigma}(x)$ is a Gaussian kernel function with zero mean and variance, σ^2 .

A combined weight is calculated for each candidate segment as the product of the individual weights described above. The top scoring candidate is selected as a seed segment for the horizon line. A greedy search is then conducted to find additional horizon line segments starting at each endpoint of the seed segment. A reference cost for the current seed segment is calculated as the distance of extending the segment to the image edge. An alternative cost is also generated for each candidate segment by calculating the distance between the seed segment and the candidate segment plus the distance of extending the candidate segment to the image edge, normalized by the reference distance. This cost is then offset by the candidate segment's weight. In this way, candidate line segments that exhibit weak visual cues serve to reinforce the path of stronger segments, while segments with strong visual cues have the ability to redirect the path of the horizon. The segment with the lowest cost solution then becomes the seed segment, and the process repeats until the image edge is reached. Fig. 4 demonstrates the major steps in constructing the horizon, while Algorithm 1 lists the calculation of the candidate segment cost.

Fig. 5 shows examples of the produced horizon line under different conditions. The first two image samples show a similar scene under differing weather conditions. In both images, a weak horizon line exists with snow-covered mountains immediately behind. This causes the horizon to visually blend with the background, which makes finding

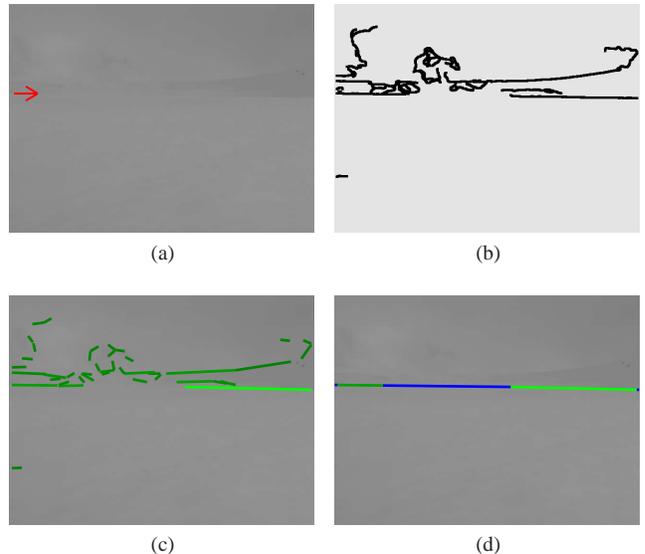


Fig. 4: (a) The sample image of Lemon Creek Glacier with the horizon line indicated by a red arrow. (b) The major edges are extracted, and (c) approximated by piecewise linear segments. Heuristic cues weight each segment, indicated by the color intensity. (d) Original segments (green) are connected to form the horizon estimate.

Algorithm 1 Candidate segment cost calculation

```

S = segments[arg max_j (segments[j].weight)]
for i = 1 to N do
  C = segments[i]
  P_0 ← (0, S.slope · 0 + S.intercept)
  P_1 ← (0, C.slope · 0 + C.intercept)
  D_ref ← ||P_0 - S.left_pt||
  D_1 ← ||P_1 - C.right_pt||
  D_2 ← ||C.right_pt - S.left_pt||
  C.cost ← (D_1 + D_2) / D_ref - C.weight
end for

```

the correct horizon line difficult, even for human observers. Despite this, the horizon line extraction process is able to identify a reasonable horizon in both images. The second two images are from the same data set, acquired several seconds apart. As the camera pans to the right, the horizon weakens to the point of becoming invisible. In fig. 5d the desired horizon has become too weak for successful detection. Despite the failure, the performance degraded gracefully, reverting to the next, stronger boundary line.

From a computation standpoint, most of the operations are applied per-segment, not per-pixel. This means that the computation time is proportional to the number of candidate line segments, not the image size. Further, most of the operations require only a histogram of a small area, which is an inexpensive calculation. The resulting algorithm is capable of running in real time, with computation times of less than 30ms on a 640x480 image.

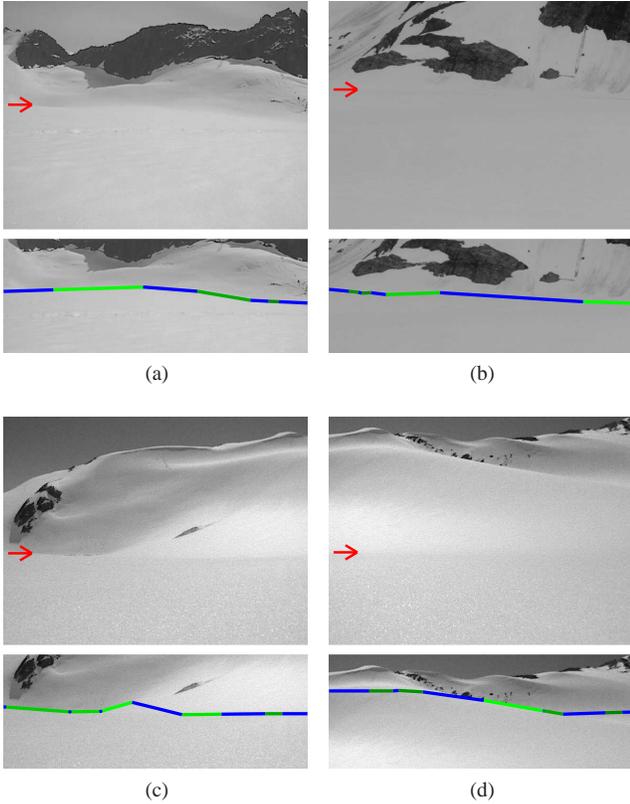


Fig. 5: Typical results of the horizon line extraction process on images acquired on Mendenhall Glacier and Lemon Creek Glacier. Top graphic shows the original image, while the bottom shows a truncated section with the extracted horizon line. The desired horizon is indicated on the original image.

V. RESULTS

To evaluate the effectiveness of the proposed region extraction algorithm, continuous segments of recorded video from each of the field trials were selected. The horizon line extraction algorithm (HL) was applied to each video segment, and the resulting region mask was recorded. To compare the results, images from each video segment have been hand labeled, indicating the area of traversable foreground. It should be noted that in some images the line between traversable foreground and background is somewhat arbitrary. An effort was made to choose a consistent line between sequential images. Ultimately, this ambiguity exists only over a small vertical range of pixels in the image, and should not unduly affect the measured performance. Due to the manpower required to hand label images, only 100 frames were selected from each video segment, uniformly spaced through time. The algorithm results are then compared to the hand labeled images, with the number of incorrectly labeled pixels counted for each frame. For comparison, the methods of adaptive histogram thresholding (AHT), region growing (RG), Gaussian mixture model (GMM), and statistical region merging (SRM), were also evaluated in the same manner. The SRM algorithm is a more advanced region growing

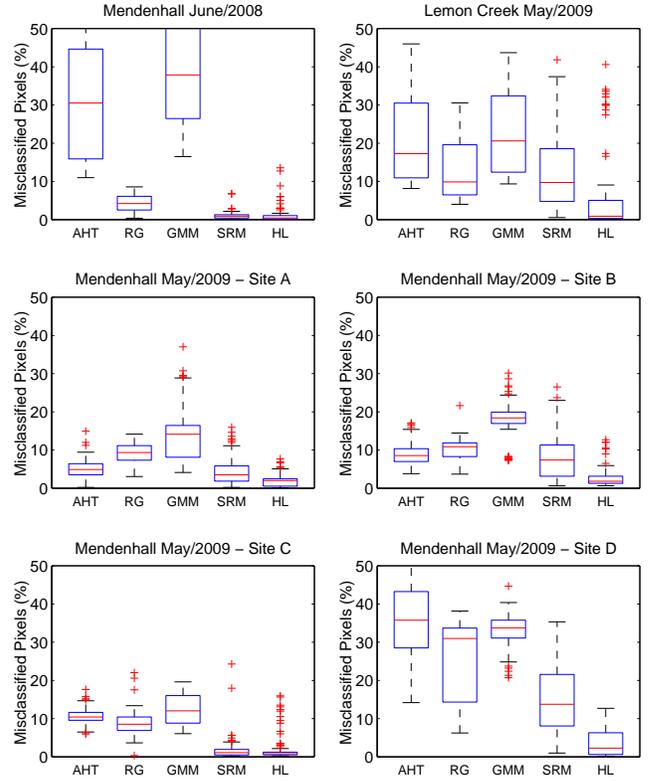


Fig. 6: Classification performance results for adaptive histogram thresholding (AHT), region growing (RG), machine learning using a Gaussian mixture model (GMM), statistical region merging (SRM), and the proposed horizon line extraction process (HL). Each algorithm was tested against 100 hand-labeled images from each of six field test locations.

operation that employs global information to produce more accurate boundary locations [8]. However, the use of global information entails performing a per-pixel sort, a numerically expensive operation. The results of each test location are summarized in Fig. 6 in the form of boxplots. Boxplots are a convenient graphical method of comparing statistical results that may not be normally distributed [16]. The box center-line indicates the median score, while the upper and lower border indicate the first and third quartile boundaries.

As illustrated in Fig. 6, AHT and GMM consistently score the worst of the methods investigated. These methods analyze each pixel without any consideration for the pixel location. Consequently, large portions of the sky tend to be misclassified. In contrast, RG and SRM only consider pixels neighboring the current region. Since these algorithms were initialized within the foreground, they are more likely to stay confined to the foreground. SRM uses global information when selecting the next candidate pixel to merge, explaining the improved performance over the standard RG algorithm.

To test the execution time of each algorithm, a single 640x480 test image was loaded into memory. Each algorithm then processed the test image 1000 times, and the elapsed processing time was recorded. The resulting average

TABLE I: Algorithm Execution Times

Algorithm	Execution Performance (Average Time Per Frame)
Adaptive Histogram Threshold	0.0074 s
Region Growing	0.0229 s
Gaussian Mixture Model	0.2677 s
Statistical Region Merging	0.2803 s
Horizon Line Extraction	0.0296 s

execution times are summarized in Table I. The adaptive histogram threshold method was the fastest, followed by region growing. The statistical region merging method, which consistently exhibited high accuracy, was only capable of operating at 3 to 4 Hz. Such low frame rate operation is unacceptable as a component of a real-time obstacle avoidance system. The Gaussian mixture model was also incapable of real-time operation. While the classification stage itself was sufficiently fast, the generation of the required feature vector was too computationally intensive. Finally, the horizon line extraction procedure was capable of greater than 20 Hz operation, making it an acceptable choice.

VI. CONCLUSIONS

A custom horizon line extraction algorithm based on visual cues was proposed. Due to the real-time, low processing requirements of field mobile robotics, special emphasis has been given to the execution time of each algorithm. The performance of each algorithm is evaluated numerically, both in terms of time and accuracy, on samples from each of six different field trials on glaciers in Alaska.

This method has been formulated to avoid the use of thresholds and tuning parameters when possible. Each visual cue is evaluated for each candidate line segment; it is not until the end of the process that candidates are culled. If the properties to which a specific visual cue responds are not present in the image, then all candidate segments get penalized a similar amount. In contrast, a threshold-based system would likely cull all of the candidates, resulting in failure. This endows the system with a certain robustness to image variation, and mitigates the effects of applying visual cues that are not optimal for a specific image.

From a classification standpoint, the proposed horizon line extraction procedure (HL) and statistical region merging (SRM) are the clear winners. In all conducted trials, the median classification error of the HL algorithm was less than 2.5% of image pixels. This result marginally outperformed statistical region merging (SRM) in every data set, but does not require the computational expense of a full pixel sort. In terms of execution speed, the HL algorithm is an order of magnitude faster than SRM, enabling real-time operation.

Currently, each cue is treated equally. However, many data fusion techniques exist for combining information from different sources [17], which can lead to more robust performance. Future work in this area will investigate the

performance benefit, if any, of incorporating an adaptive data fusion system into the horizon classification process. Additionally, this system operates on raw images. Recent work in contrast enhancement and dehazing [18] may serve as a useful preprocessing step, particularly in overcast conditions.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to our collaborators Dr. Derrick Lampkin, Pennsylvania State University, for providing the scientific motivation for this research, Dr. Magnus Egerstedt, Georgia Institute of Technology, for providing his experience in multi-agent formations, and Dr. Matt Heavner, Associate Professor of Physics, University of Alaska Southeast, for providing his expertise in glacial field work.

REFERENCES

- [1] S. Williams and A. M. Howard, "Towards visual arctic terrain assessment," in *International Conference on Field and Service Robotics, FSR*, Cambridge, MA, July 2009.
- [2] —, "Developing monocular visual pose estimation for arctic environments," *Journal of Field Robotics*, vol. 27, no. 2, pp. 145–157, 2010.
- [3] N. Pears and B. Liang, "Ground plane segmentation for mobile robot visual navigation," in *IROS 2001*, vol. 3, 2001, pp. 1513–1518.
- [4] Y. geun Kim and H. Kim, "Layered ground floor detection for vision-based mobile robot navigation," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 1, 26 2004, pp. 13–18.
- [5] S. Etinger, M. Nechyba, P. Ifju, and M. Waszak, "Towards flight autonomy: Vision-based horizon detection for micro air vehicles," in *Florida Conference on Recent Advances in Robotics*, vol. 2002. Citeseer, 2002.
- [6] A. Broggi and A. Fascioli, "Artificial vision in extreme environments for snowcat tracks detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, pp. 162–172, Sept 2002.
- [7] M. Manz, M. Himmelsbach, T. Luettel, and H. Wuensche, "Fusing LIDAR and Vision for Autonomous Dirt Road Following," *Autonome Mobile Systeme 2009*, pp. 17–24, 2009.
- [8] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1452–1458, 2004.
- [9] J. Zhang and H.-H. Nagel, "Texture-based segmentation of road images," in *Intelligent Vehicles Symposium*, Paris, France, Oct 1994, pp. 260–265.
- [10] A. Saxena, S. Chung, and A. Ng, "Learning depth from single monocular images," *Advances in Neural Information Processing Systems*, vol. 18, p. 1161, 2006.
- [11] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, "Selfsupervised monocular road detection in desert terrain," in *Proc. of Robotics: Science and Systems (RSS)*. The MIT Press, 2006, p. 33.
- [12] S. Fefilatyev, V. Smarodzinava, L. O. Hall, and D. B. Goldgof, "Horizon detection using machine learning techniques," in *Machine Learning and Applications, 2006. ICMLA '06. 5th International Conference on*, Dec 2006, pp. 17–21.
- [13] I. Ulrich and I. Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," in *National Conference on Artificial Intelligence*, Austin, TX, Aug 2000, pp. 866–871.
- [14] E. Davies, *Machine vision*, 3rd ed. Morgan Kaufmann, 2005.
- [15] J. Hershberger and J. Snoeyink, "Speeding up the Douglas-Peucker line simplification algorithm," in *Proceedings of the 5th International Symposium on Spatial Data Handling*, vol. 1, 1992, pp. 134–143.
- [16] R. McGill, J. Tukey, and W. Larsen, "Variations of box plots," *American Statistician*, pp. 12–16, 1978.
- [17] M. Spengler and B. Schiele, "Towards robust multi-cue integration for visual tracking," *Machine Vision and Applications*, vol. 14, no. 1, pp. 50–58, 2003.
- [18] J. Tarel and N. Hautiere, "Fast visibility restoration from a single color or gray level image," in *IEEE International Conference on Computer Vision*. IEEE, 2010, pp. 2201–2208.