# History-Aware Free Space Detection for Efficient Autonomous Exploration using Aerial Robots

**Ryan Fite**
**Colorado School of Mines**
**Golden, CO 80401**
**425-516-5994**
**ryanfite@live.com**

**Shehryar Khattak**
**Autonomous Robots Lab**
**University of Nevada, Reno**
**Reno, NV 89557**
**shehryar@nevada.unr.edu**

**David Feil-Seifer**
**Robotics Research Lab**
**University of Nevada, Reno**
**Reno, NV 89557**
**dave@cse.unr.edu**

**Kostas Alexis**
**Autonomous Robots Lab**
**University of Nevada, Reno**
**Reno, NV 89557**
**kalexis@unr.edu**

***Abstract*— In this work, we present an approach for the detection of the direction of free space in order to improve the efficiency of robotic exploration by exploiting the history of free space calculations. As a motivational example, we consider the case of exploration of subterranean environments where the length of corridors can exceed the range of most sensors, multi–branched geometry may lead to ambiguity with respect to the most efficient direction of exploration, or sensor degradation can shorten the effective depth range. The proposed method can be used to assist a path planner by determining the directions of probable free space for efficient exploration. The algorithm was evaluated using point clouds from two types of sensors, namely sparse long-range sensors such as a LiDAR and dense short-range sensors such as direct depth RGBD sensors. Furthermore, evaluation took place against a variety of environments using handheld and aerial robotic data in urban and subterranean environments. During each of the tests, the algorithm has shown to be capable of consistently and reliably finding the directions of probable unobserved free space in real–time. As a final evaluation step, the proposed algorithm was integrated as part of the path planning functionality on–board an autonomous aerial robot and the relevant mine exploration field results are shown. Analysis of computational efficiency is further presented. The code for this method is open–sourced and accompanies this paper submission.**

## TABLE OF CONTENTS



## 1. INTRODUCTION

Underground exploration and mine management are essential tasks in the mining and drilling industries, and for most of history, these tasks are performed by humans. Relevant activities include those of mapping, rescue operations, and financial tracking of volumes of ore. The recent development of unmanned aerial vehicles provides an opportunity to perform such tasks autonomously with the benefit of reducing costs and the risk to human life [1–11]. The development of autonomous drones capable of navigating mines and cave systems would allow for the exploration and mapping of tunnels [12, 13] without having to rely on humans. A crucial part of this development is the creation of methods that enable aerial robots to map and navigate in mines accurately. Autonomous exploration of underground environments is particularly challenging because mines are often geometrically self–similar and relatively textureless environments with low visibility. In addition to this, the aerial robots have no access to the Global Positioning System meaning they have to orient and map the drifts, heads and other cave/network–like structures themselves without external assistance.

Current mapping techniques used on aerial robots deployed in underground environments rely on sensor readings to navigate the environment and to determine the areas in which the drone can and cannot travel. While these navigation techniques are generally sufficient, many cave/mine settings contain long corridors and tunnels that render the robot's on–board sensing to become unable to receive complete information due to the end of the shaft being outside of sensor range.

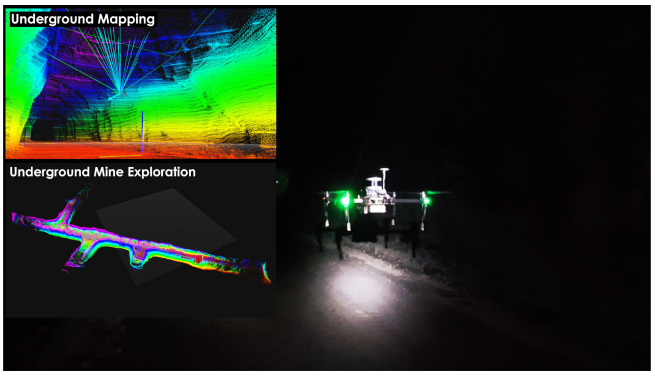


**Figure 1**. LiDAR–based mapping of an underground mine using an aerial robot. Upper left: Perspective view of underground mapping. Lower Left: Overview of explored area. Right: Drone flying in reduced visual conditions.

Without sensor returns the aerial robot is unable to perceive the area going down the corridor as free space. An example of this problem is shown in Figure 1, where the LiDAR scan is incapable of mapping all the way down the mine corridor resulting in blank space in the created map. Without the ability to map the passage as an area of obstacle–free space, the path planner on the robot has trouble efficiently

978-1-5386-6854-2/19/$31.00 ©2019 IEEE

navigating the mine corridor. To overcome this issue, we present an approach for improving the efficiency of exploration by exploiting the history of free space calculations to detect directions of free space when sensors are limited by the environment. Sensor limitations, like those encountered in subterranean environments, can lead to a reduction in the robots capacity to explore efficiently. The system tends to sample somewhat randomly resulting in inefficient exploratory motion, in an attempt to safely observe free space while exploring (given sensor limitations), as opposed to traveling down the passageways center. Traversing down the center of the passageway would greatly improve the amount of exploration that a robot with a limited payload - such as a Micro Aerial Vehicle (MAV) - can accomplish within its operational time budget. The method also distinguishes between gaps in sensor readings and the actual regions of open space using the local environment of the aerial robot to avoid issues caused by drift in the overall mapping. This method draws inspiration from the vector field histogram method [14] and its several variants. It can be used to assist the aerial robot's navigation and exploratory behavior by creating these direction vectors as guides for travel. In order to comprehensively evaluate the proposed method, results based on both a) dense short–range depth sensor observations in urban environments, as well as b) sparse long–range LiDAR sensor observations inside an underground mine are presented. As a final evaluation step, the proposed algorithm was integrated as part of the path planning functionality on–board an autonomous aerial robot [3] and relevant mine exploration field results are shown. Analysis of computational efficiency is further presented. The code for this method is open–sourced and available at `https://github.com/unr-arl/hfsd`, while a video of relevant experimental results can be found at `https://www.autonomousrobotslab.com/hfsd.html`

This paper is organized as follows. Section 2 discusses work and prior contributions related to the proposed approach. The derived algorithm is detailed in Section 3, while experimental evaluation studies are presented in Section 4. Finally, conclusions are drawn in Section 5.

## 2. Related Work

The work in this paper is inspired by the iterations of the Vector Field Histogram (VFH) method. The following section will briefly describe each iteration of the method and its effect on the method presented in this paper.

### *The VFH Method*

The VFH method is a two–step data reduction method originally designed to be used for ground robots with sonar sensors. The method focuses on fast obstacle avoidance based on a temporary local map of the robot's environment. The VFH method works by continually updating the robot's environment using ranging data and defining an active view window of $W \times W$ cells for assigning magnitudes of polar obstacle density to each cell. The magnitude of the cells are then used to construct a one–dimensional polar histogram with bins described by a section of the azimuth around the robot. The polar histogram is the core contribution of the VFH method. The data is then smoothed to improve accuracy by accounting for possible errors in the histogram grid. The algorithm chooses candidate valleys from lengths of sectors in the polar histogram that lie below a safety threshold and selects the candidate valley with directions closest to the robot's target. The robot is then commanded to steer towards the sector in the middle of the valley with a velocity dependent on the polar object density [14]. The advantage of the VFH method over previous methods is that although it maintains an overall representation of its surroundings in an on–board map it only takes into account its immediate surroundings for obstacle avoidance as opposed to previous methods that required the map of the whole environment for obstacle avoidance without regarding the locality of objects. The main flaws with respect to this method relate to the fact that it does not account for the width of the robot and uses only a single threshold to determine candidate valleys which can be volatile if set too high or too low. An inefficient form of path planning is also used due to the lack of an ability to look ahead in the environment. The proposed method, among others, draws motivation from the original VFH but emphasizes on resolving its limitations and extending its scope.

### *The VFH+ Method*

The VFH+ method advanced the original VFH method by addressing some of the most serious flaws within the VFH method. The significant contributions of the VFH+ method include considerations for the size of the robot, a hysteresis based threshold system instead of a single threshold, considerations for the dynamics and kinematics of the robot, and the creation of a cost function to improve calculation of the proper steering direction [15]. The most prominent advancement that is not robot specific is the addition of the cost function based on the difference between the candidate direction and the direction of the robot's target i.e. the robot's current wheel orientation and previously chosen direction respectively. The cost function allows for the algorithm to choose the best candidate direction based on the immediate consequences of a certain movement. These additions allowed the robot to better navigate the environment and to select goal–oriented paths. The limitation that remains in this method is that it is too local and can cause the robot to get stuck in dead ends of the environment.

### *The VFH$^\star$ Method*

The VFH$^\star$ method is the current iteration of this method combining the VFH+ method with the A$^\star$ search algorithm to create a local obstacle avoidance algorithm with the capability to look ahead using search trees. The primary issue that the VFH$^\star$ method addressed was the inability for purely local obstacle avoidance algorithms to accurately choose the best path. Purely local algorithms often fail because they only consider the immediate consequences of their next move within their defined sample range which can result in algorithms choosing directions that lead into dead ends [16]. The VFH$^\star$ algorithm addresses this issue by projecting the candidate paths several steps ahead to find the path with the lowest cost. This is accomplished by repeating the VFH+ algorithm iteratively along candidate trajectories to create a search tree with nodes containing the cost of trajectories. The algorithm repeats this process until the nodes hit a specified distance away from the starting position then using a pre–order traversal of the tree and the sum of the cost function of each node for the root's primary candidate directions is found and evaluated heuristically. The search tree allows for basic on–board path planning in a local environment. The primary advantage of this method is the ability to look ahead and weigh the consequences of movement in the primary candidate directions, while maintaining both a high computational and physical speed for the robot. The idea that this method inspired was the history window. In further detail, this method utilizes a look ahead verification paradigm which

naturally cannot be used inside unknown environments. In the proposed approach we utilize a look behind method that helps to evaluate trajectories based on the most probable shape of the environment according to past measurements.

## 3. PROPOSED METHOD

The method presented in this paper is a robotic perception algorithm that uses a history of previous depth/range measurements to find appropriate directions of exploration for robots operating in an unknown environment. The overview of this method is shown in Figure 2.

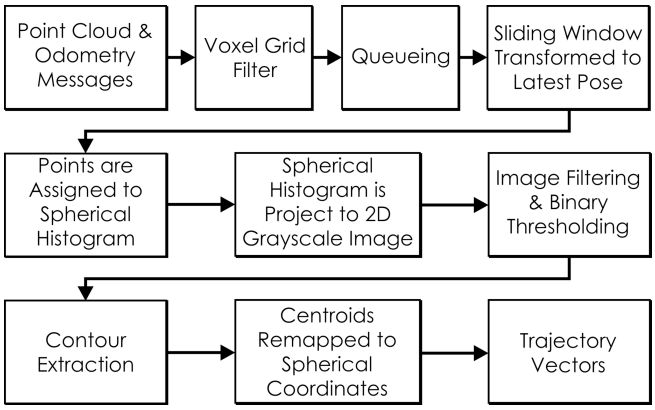


**Figure 2**. Overview of the steps between the input of odometry and point cloud messages to the resulting trajectory vectors outputted by the algorithm.

The measurements may come from the use of a 3D LiDAR, an RGBD camera or any other depth sensor. This method is capable of proposing appropriate directions for exploration by utilizing a sliding–window history of the robots pose estimates and the depth measurements of the environment. More specifically, the method finds areas of sparse sensor returns near the end of the robots perception and determines the direction to these areas as the probable directions of free space due to the consistency of sensor readings with the shape of the environment. In tunnel–like environments, the probable direction of free space is likely to be from the center of the tunnel to its exit, while in other cases it may be towards the center of an open room. This method can be used to assist a path planner by determining the directions of probable free space for efficient exploration. In practical terms, the algorithm utilizes odometry estimates, and Point Cloud measurements in order to output travel vector proposals in directions of unexplored space. Any path planning module can then exploit this information towards more efficient exploration.

*Preprocessing and Queueing*

The algorithm pulls in odometry and point cloud measurements and converts the point cloud to the world frame ($\mathcal{W}$) from the robot frame ($\mathcal{R}$) so that it can be aligned to the current pose later in the algorithm operation. This transformation is done by finding the affine transform matrix that is able to convert sets of point clouds from their current odometry expressed in $W$ to the associated $R$ frame. A voxel grid filter is then applied to the most recent point cloud measurements in order to reduce the number of points to speed up transformation to the sliding window.

The point clouds are then added to a queue with a variable maximum queue size. The point clouds are queued in this manner to allow for the algorithm to exploit the sensor history by using a combination of both its currently visible data and past data. A second queue is employed to maintain the original pose data for later transformation. To remove the effect of odometry drift, which could impede the accuracy of our method, all point clouds are aligned to the most recent pose as opposed to being aligned to the base world pose. This is done because when aligning with the base world pose small inaccuracies between consecutive pose measurements become compounded and drift from the actual position of the points becomes problematic. Conversely, by transforming the point clouds to the latest pose the compound drift effect is lessened because the error accumulated in measurements before the section of history maintained in the algorithm is discarded.

*Image Projection*

Once point clouds have been aligned, this local map of the environment is projected to a matrix that represents the spherical relationship between the robot and the points in the surrounding point clouds. The relationship between the closest points to the robot and the robots position in the current history is used to create a spherical proximity matrix. The environment is then represented as a spherical histogram that is divided into $v \times h$ sectors. Each sector represents a specific number of degrees along the azimuth ($h$) or elevation ($v$). The radius of a point in the point cloud that is closest to the robot within a given sector is then assigned to that sector in a matrix. If no results are contained within that spherical sector, it remains at the lowest value. This is done to ensure the robot's safety when there are no returns in case the lack of returns is caused by a sensor failure as opposed to being open space. In addition to this safeguard, a minimum distance rejection filter immediately drops any sectors with a radius below a specified threshold to zero. Each matrix entry is then assigned a grayscale intensity according to Equation 1.

$$I = \min((255 - O) \cdot (R/M)^2, 255) \quad (1)$$

In this equation the intensity ($I$) is equal to the lower value between 255, which is pure white meaning extremely far away, and 255 minus the offset ($O$) which is an adjustable value to raise or lower the sensitivity multiplied by the current sector's radius ($R$) divided by the largest sector radius in all of the sectors ($M$) squared. High–intensity values represent sectors that have the closest point that is relatively far away from the robot. In a typical hallway, this image could resemble a ring of white space around an area of entirely black space which represents the floor, walls, and ceiling at the edge of the robot's sensor range. An example of this is shown in Figure 3.

*Image Filtering and Thresholding*

The derived grayscale image is then dilated to emphasize the points that are far away from the robot. The image is then put through a Gaussian blur filter to limit the effect of sensor errors. This promotes consistency between the data and the actual environment especially in areas at the edge of the robot's vision because it makes the inference that the shape of the surroundings is relatively consistent. The grayscale image is subsequently converted into a binary image using an Otsu threshold[17]. This type of threshold allows for adaptive thresholding of 2D images so that the method is capable of adapting to a variety of operating environments.

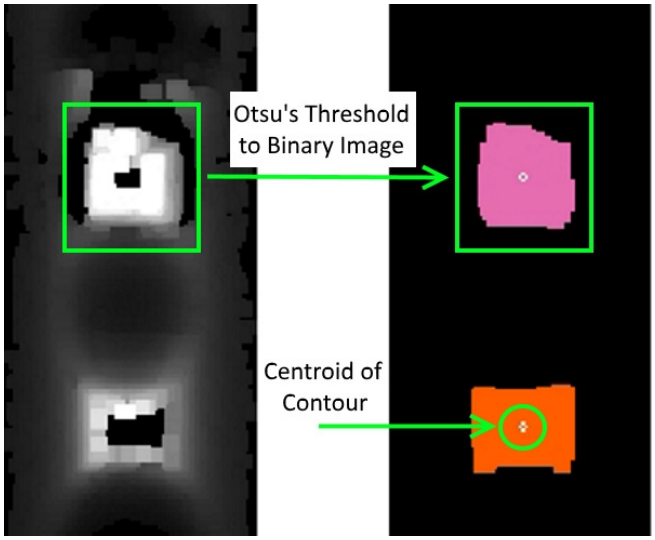


**Figure 3**. Left: The grayscale image after it has gone through the Gaussian blur and has been dilated. Right: The extracted contours after the us of an Otsu threshold (The white portions of the binary image have been colored in for differentiation). The associated centroids of the contours are shown in white.

*Contour Extraction and Direction Vectors*

The contours of the image are then extracted and separated. The contour extraction and image thresholding are shown in Figure 3. The centroids of these contours are used to describe a proposed direction of travel for optimized exploratory operation. The pixel values of the centroids are converted back to spherical coordinates. These directions are assigned a radius value equal to the size of the contour to maintain a metric of the amount of exploration possible in the direction of the proposed vector. Finally, the proposed vectors are converted to Cartesian coordinates in the world frame to be used by the aerial robot.

## 4. Experimental Evaluation

The proposed method was evaluated in both urban indoor environments and underground mine corridors. Our primary focus with respect to operating in these environments was to test the algorithm's capabilities in corridors with varying forms of geometry as expected in many environments of operational significance such as mines, subways and tunnels. Urban indoor environments have many flat surfaces and right angles while the subterranean corridors have much more ambiguous geometry with few flat surfaces and diverging branches at odd angles.

We tested two primary sensing modules: a dense short range Picoflexx Monstar RGB–D camera, and a long range sparse depth LiDAR Velodyne PuckLITE. The Picoflexx Monstar only has a range of $0.5$ to $6$ meters so it is only appropriate for short range sensing. The PuckLITE LiDAR has a $360°$ horizontal field of view, a $30°$ vertical field of view, and a range of up to $100$ meters in good conditions, while running at up to $20$ Hz making it ideal for long range sensing. The complete relevant specifications of these sensors are shown in Table 1.

| Sensor | Picoflexx Monstar | Velodyne PuckLITE |
|---|---|---|
| Range | 0.5 – 6 meters | 100 meters |
| Resolution | $352 \times 287$ pixels | $2° \times 0.1° - 0.4°$ |
| Field–of–View | $100° \times 85°$ | $360° \times 30°$ |
| FPS/Frequency | 60 FPS | 20 Hz |
| Weight | 142 grams | 590 grams |

**Table 1**. Specifications of sensors used in the experiments.

*Indoor Evaluation*

We evaluated the algorithm on an indoor urban environment using the Monstar sensor. We used the Monstar because the environment was close quarters so we could rely and test the limited range of the Monstar. The purpose of this experiment was to evaluate the consistency of the algorithm in an environment that has no divergent paths and has primarily flat surfaces in order to show the effect of history and voxel grid size in a geometrically consistent environment. The geometric consistency of the environment allowed us to primarily evaluate the effect of history and voxel grid size in a much more controlled environment than those that exist in underground tunnels.

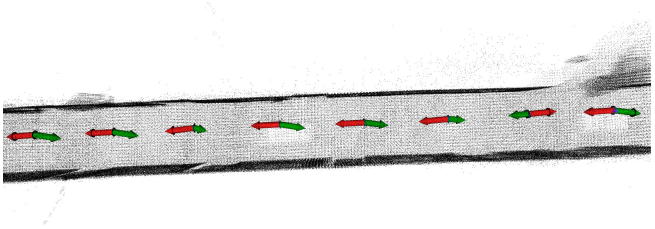

**Figure 4**. Map of the straight hallway recorded using a Picoflexx Monstar RGB-D camera with generated directional vectors by the proposed approach.

The map of this test is shown in Figure 4. We evaluated the consistency of our algorithm in this environment by evaluating the deviation from the mean of the generated directional vectors pointing along the direction of the corridor.

In particular, we first evaluated the effect of changing the amount of measurement history on the consistency of the algorithm. We ran four tests with different lengths of history available to the algorithm, namely using $[10, 40, 70, 100]$ past measurements as depicted in Figure 5. When evaluating the consistency of the algorithm against a selected window length, the Monstar-based data presented a reduction in deviation from the mean as the length of history increased which is shown in Figure 5. In this Figure the deviation in degrees is the angle between the measured vector to the mean vector for a set of directional unit vectors.

The reduction in deviation from the mean due to history length is expected because a longer sliding window of observations assists in resolving the limitations of the Monstar and specifically the fact that the Monstar has a short range and it is a directional camera. By using a long history of measurements with the Monstar, the algorithm can exploit information from both a) a further distance than the camera's actual current range measurements and b) in directions the camera is not currently viewing by utilizing a window of the sensor's measurement history. By being able to account for a history of environment observations, the consistency of the algorithm is significantly improved.

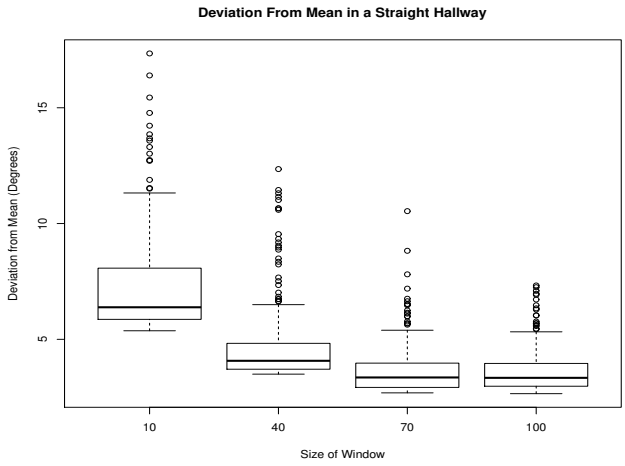


**Figure 5**. Consistency of vectors compared to the mean in a long straight corridor based on different window lengths. Consistency is based on the total degrees of the arc between the a unit vector in the set and the mean of the set.

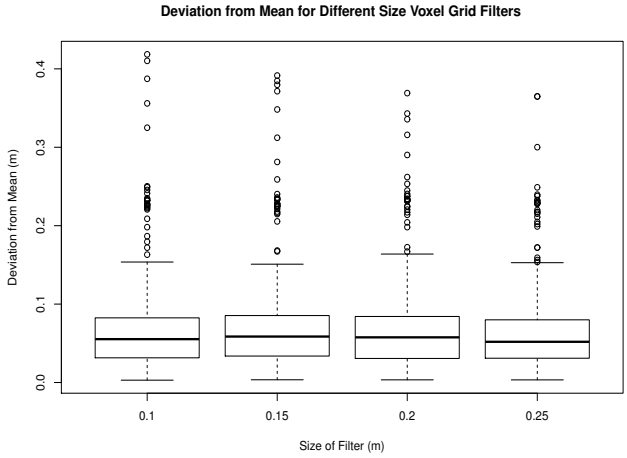


**Figure 6**. Consistency of vectors compared to their mean in a long straight corridor based on different voxel grid sizes. Consistency is based on the total degrees of the arc between the a unit vector in the set and the mean of the set.

The algorithm's consistency was then evaluated based on the size of the voxel grid filter. The purpose of evaluating this is to test the effect the voxel grid has on consistency due to the removal of excess points. The less points the algorithm has to process the faster it executes, but it is important to ensure that the removal of points using a voxel grid filter would not impact in any significant manner the consistency of the result. The evaluation of the voxel grid is shown in Figure 6 and the window size evaluated for this voxel grid was a window size of $40$. As the plot depicts, the consistency of the algorithm is not heavily affected by the size of the voxel grid filter. This is due to the algorithm not requiring many points occupying the same space due to the spherical histogram only needing the value of the closest point in a sector of the robot's environment. This in turn means that even with the removal of a significant amount of points the consistency is not heavily effected.

*Underground Evaluation*

We further evaluated our algorithm in an underground mine environment. The subterranean corridor evaluation was conducted with the LiDAR PuckLITE. We used the PuckLITE LiDAR because it has a long range and a $360°$field of view. The range and field of view of the PuckLITE is useful in this environment as the corridors are much longer and there are diverging paths. The purpose of this evaluation was to test the consistency of the algorithm in an environment with divergent paths and very few flat surfaces. This test case was designed to specifically evaluate the effect of multi-branched and ambiguous environmental geometry on the algorithm's consistency. In addition to the evaluation's primary purpose the execution time of the algorithm was also evaluated during this experiment.

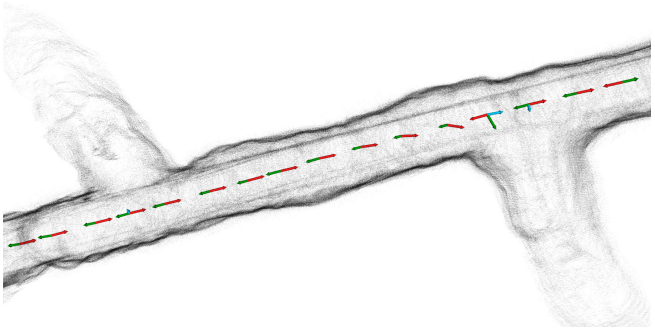

**Figure 7**. Map of the hallway with diverging paths recorded using a long range sparse depth LiDAR PuckLITE sensor with generated directional vectors based on our approach.

The map of this test is shown in Figure 7. The consistency of the algorithm was evaluated in the same manner as the indoor evaluation. Specifically, we evaluated the effect of changing the amount of measurement history on the consistency of the algorithm. We ran four tests with different lengths of history available to the algorithm, namely a window of $[10, 40, 70, 100]$ observations as depicted in Figure 8. When evaluating the consistency of the algorithm based on window length, the result using the PuckLITE had less deviation from the mean as the effective length of window history increased.

The results of this evaluation are shown in Figure 8. The derived result is expected because when more history of the straight portions of the corridor are taken into account the branching paths and inconsistent walls of the corridors have less effect on the consistency of the directional vectors outputted by the algorithm. This is because the PuckLITE has sparse measurements and cannot view outside of a $30°$vertical field of view centered at its horizontal. By using measurement history with the PuckLITE, the algorithm has more dense information due to repeated measurements and the previous measurements in the history window fill the gaps of information outside of the LiDAR's $30°$vertical field of vision. In comparing the history based results between the PuckLITE and the Monstar we attribute the difference between the amounts of deviation with a low window size to the differences in range and directional data between the two sensors. However, the reduction in deviation away from the mean vector as the length of the measurement history window increases is consistent with both types of sensors in their respective appropriate environments.

Finally, in addition to evaluating the consistency of the algo-

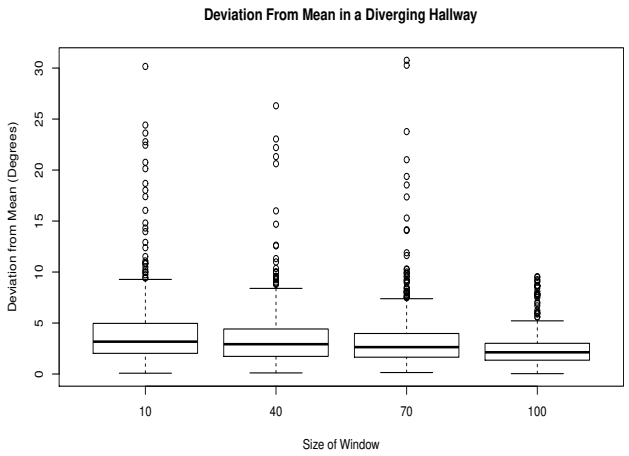


**Figure 8**. Consistency of vectors compared to the mean in a long corridor with diverging paths based on different window lengths. Consistency is based on the total degrees of the arc between the a unit vector in the set and the mean of the set.

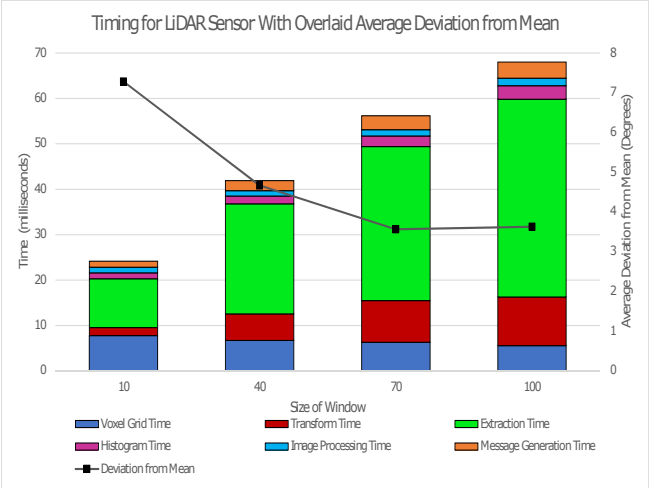


**Figure 9**. Timing breakdown of the algorithm based on a voxel grid size of 0.2m for various window sizes using the LiDAR PuckLITE. Each colored bar represents a different major step in the algorithm's process. Additonally a line has been added depicting the average deviation from the mean at the different window sizes to present the time-accuracy trade-off.

rithm within this mine corridor we also evaluated the amount of time the algorithm takes to complete each step within its methodology. The results of this timing evaluation are shown in Figure 9. The execution time of the algorithm is most effected by the number of points that have to be extracted from the robot's environment and secondarily by the number of clouds that have to be aligned. This timing data was taken with the LiDAR PuckLITE which ran at a rate of 10Hz meaning that the algorithm is capable of running in real time even with a history of the robot's last ten seconds of measurement.

*Autonomous Underground Mine Exploration*

As a final step, the proposed history–aware free space detection algorithm was combined with the previously proposed receding horizon next-best-view-planner (NBVP) planner [3] in order to enable improved autonomous exploration in underground mines. In particular, the modified NBVP planner utilizes the direction proposals from the algorithm outlined in this paper to focus its sampling process in the areas along the directions of maximum free space history. To bias this sampling process, we replace the uniform sampling strategy employed in the original NBVP implementation (open–sourced at: `https://github.com/ethz-asl/nbvplanner`) with a set of normal distributions the centers of which are along the directions of the vectors derived by the method described in this paper. For the experimental results, a micro aerial vehicle equipped with a PuckLITE is utilized while its on-board localization and mapping is achieved through the use of the LiDAR Odometry And Mapping (LOAM) algorithm [18]. Figure 10 presents the relevant results. In particular, we verified autonomous exploration supported by this algorithm both in a part of an underground mine that involved branching to different directions, and across a part of a mine drift in which a clear direction of maximum exploration efficiency exists.

## 5. Conclusions

In this paper we presented a method to improve the efficiency of autonomous exploration by taking an input of odometry and point cloud messages. The method's evaluation was done on a variety of environments and using two sensing modules. The method has shown to be consistent in its operation on both sensing modules in their appropriate environments. The primary advantage that this method provides is that it reduces the randomness of sampling during exploration. The algorithm can be used to assist a path planner by determining the directions of probable free space for efficient exploration. The reduction of excess sampling in corridors is a vast improvement to the efficiency of exploration in subterranean environments. By making it easier for a path planner to explore environments this method enhances the capability of robots with limited payload to explore large unknown environments within their operating time.

## Acknowledgments

This work was supported by a) the National Science Foundation (IIS-1757929) and b) the Mine Inspection Robotics project sponsored by the Nevada Knowledge Fund administered by the Governor's Office of Economic Development.

## References

[1] C. Kanellakis and G. Nikolakopoulos, "Evaluation of visual localization systems in underground mining," in *Control and Automation (MED), 2016 24th Mediterranean Conference on*. IEEE, 2016, pp. 539–544.

[2] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty–aware receding horizon exploration and mapping using aerial robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2017. [Online]. Available: https://github.com/unr-arl/rhem_planner

[3] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.

[4] C. Papachristos, F. Mascarich, and K. Alexis, "Thermal-

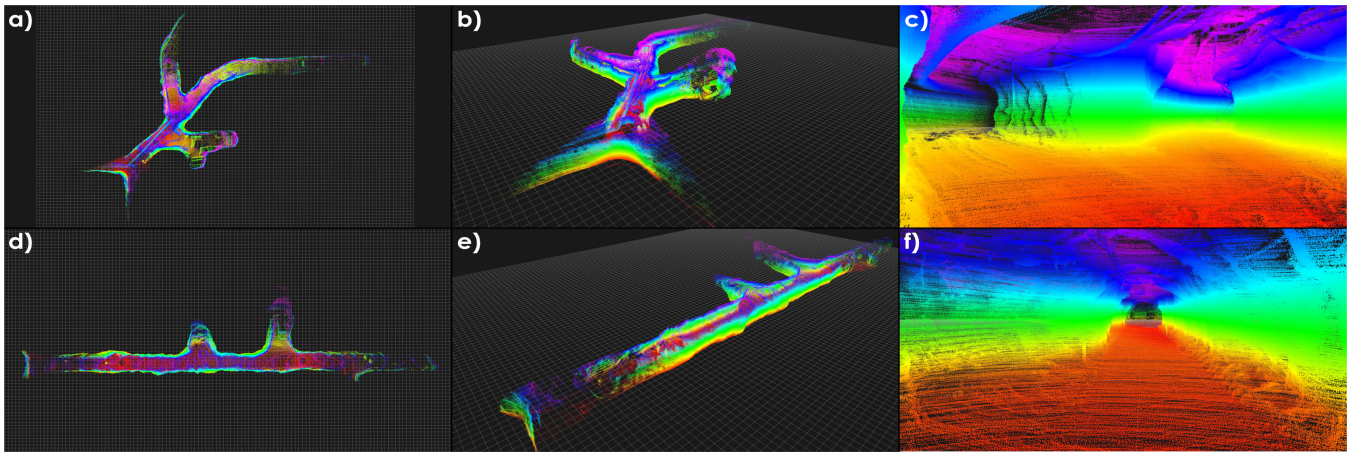


**Figure 10**. Results of autonomous aerial robotic exploration inside an underground mine using a modification of the previously published and open–sourced NBVP planner such that it accounts for the vector proposals of the algorithm presented in this paper. A micro aerial vehicle equipped with a PuckLITE is exploring an underground mine and was tested both in a part of the mine involving branching to multiple directions (upper row) and a part of a mine drift that has a clear direction of progression (lower row). Top-view [a), d)], oblique-view [b), e)] and detailed through–the–map views [c), f)] are presented.

inertial localization for autonomous navigation of aerial robots through obscurants," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 394–399.

[5] S. Khattak, C. Papachristos, and K. Alexis, "Marker based thermal-inertial localization for aerial robots in obscurant filled environments," in *International Symposium on Visual Computing*. Springer, 2018, pp. 565–575.

[6] T. Dang, C. Papachristos, and K. Alexis, "Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[7] S. Khattak, C. Papachristos, and K. Alexis, "Vision-depth landmarks and inertial fusion for navigation in degraded visual environments," in *International Symposium on Visual Computing*. Springer, 2018, pp. 529–540.

[8] T. Dang, S. Khattak, C. Papachristos, and K. Alexis, "Visual-inertial odometry-enhanced geometrically stable icp for mapping applications using aerial robots," *arXiv preprint arXiv:1801.08228*, 2018.

[9] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Autonomous Robots*, pp. 1–25, 2015.

[10] C. Papachristos, M. Kamel, M. Popović, S. Khattak, A. Bircher, H. Oleynikova, T. Dang, F. Mascarich, K. Alexis, and R. Siegwart, "Autonomous exploration and inspection path planning for aerial robots using the robot operating system," in *Robot Operating System (ROS)*. Springer, 2019, pp. 67–111.

[11] S. Khattak, C. Papachristos, and K. Alexis, "Change detection and object recognition using aerial robots," in *International Symposium on Visual Computing*. Springer, 2016, pp. 582–592.

[12] F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "A multi-modal mapping unit for autonomous exploration and mapping of underground tunnels," in *IEEE Aerospace Conference (AeroConf)*, March 2018.

[13] C. Papachristos, S. Khattak and K. Alexis, "Autonomous exploration of visually-degraded environments using aerial robots," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2017.

[14] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE transactions on robotics and automation*.

[15] I. Ulrich and J. Borenstein, "Vfh+: Reliable obstacle avoidance for fast mobile robots," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2. IEEE, 1998, pp. 1572–1577.

[16] I.Ulrich and J.Borenstein, "Vfh*: Local obstacle avoidance with look-ahead verification," in *ICRA*.

[17] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[18] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, 2014, p. 9.

## BIOGRAPHY

***Ryan Fite*** *is a student currently pursuing his B.S. in Computer Science at Colorado School of Mines. As of 2018, he is working as a researcher at the Mines Interactive Robotics Research Lab, a human robot interaction research lab at Colorado School of Mines. He is also working as a community outreach representative to teach kindergarten through high school students about robotics.*

***Shehryar Khattak*** *earned his B.S. in Mechanical Engineering from Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan in 2009 and M.S. in Aerospace Engineering from Korea Advanced Institute of Science and Technology, Daejeon in 2012. From August 2012 to December 2015, he worked as a Research Engineer at Samsung Electronics in Suwon, South Korea. Currently, Shehryar is pursuing his Ph.D. in Computer Science and Engineering from the University of Nevada, Reno. His current research is related to robot perception and path planning with focus on development of localization and mapping algorithms exploiting multi-sensor information.*

***David Feil-Seifer*** *received the B.S. degree in computer science from the University of Rochester, Rochester, NY, USA, in 2003 and the M.S. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, CA, USA, in 2007 and 2012, respectively. He has been an Assistant Professor and the Director of the Socially Assistive Robotics Group, Department of Computer Science and Engineering, University of Nevada, Reno, NV, USA, since 2013. From 2011 to 2013, he was a Post-Doctoral Associate with the Computer Science Department, Yale University, New Haven, CT, USA.He has authored over 45 papers published in major journals, book chapters, and international conference proceedings. His current research interests include human-robot interaction and socially assistive robotics. Dr. Feil-Seifer is the Managing Editor of the ACM Transactions on Human-Robot Interaction.*

***Kostas Alexis*** *obtained his Ph.D. in aerial robotics control and collaboration from the University of Patras, Greece in 2011. From 2011 to 2015 he held the position of Senior Researcher at the Autonomous Systems Lab of ETH Zurich. Currently, Dr. Alexis is an Assistant Professor at the University of Nevada, Reno and director of the Autonomous Robots Lab. His research interests lie in the fields of robotics and autonomy. He is the author of more than 70 publications and has received multiple best paper awards.*