

# AUTOMATIC TRAJECTORY GENERATION FOR AUTONOMOUS ROBOTIC SHOTCRETING IN UNDERGROUND MINES

Michael R. Wrock<sup>1</sup>, Scott B. Nokleby<sup>1</sup>

<sup>1</sup>*Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, Oshawa, Canada  
Email: michael.wrock@uoit.ca; scott.nokleby@uoit.ca*

---

## ABSTRACT

A trajectory generation algorithm for an autonomous robotic system performing the task of spraying shotcrete (sprayable concrete) in underground mines is presented. The main requirement for shotcreting is to orient the end-effector perpendicular to the mine surface. As well, the manipulator trajectory must maintain an optimal distance from the mine surface and not miss any portion of the selected area. A LiDAR (Light Detection and Ranging) scanner on a nodding head produces point clouds that are used as the input for the trajectory generation algorithm. The algorithm then generates a set of position and orientation via points that the manipulator must follow in order to perform the shotcreting task. The developed algorithm has been successfully tested in a scaled mock-up of an underground mine.

**Keywords:** trajectory generation, autonomous, shotcrete, underground mining.

---

## GÉNÉRATION AUTOMATIQUE DE LA TRAJECTOIRE D'UN ROBOT POUR LA PROJECTION DE BÉTON DANS LES MINES SOUTERRAINES

### RÉSUMÉ

Ce papier traite d'un algorithme de génération de trajectoire pour un système robotique autonome chargé d'arroser du béton projeté (béton propulsé) dans des mines souterraines. La principale exigence en matière de projection est d'orienter l'effecteur robotique perpendiculairement à la surface de la mine. De plus, la trajectoire du manipulateur doit rester à une distance optimale de la surface de la mine et ne manquer aucune partie de la zone sélectionnée. Un scanner LiDAR (détection et estimation de la distance par la lumière) sur une tête inclinée produit des nuages de points qui sont utilisés comme entrées pour l'algorithme de génération de trajectoire. L'algorithme génère ensuite un ensemble de positions et d'orientations via des points que le manipulateur doit suivre pour effectuer la tâche d'arrosage. L'algorithme développé fut testé avec succès dans une maquette à l'échelle d'une mine souterraine.

**Mots-clés :** génération de trajectoire, autonome, béton projeté, exploitation minière.

# 1. INTRODUCTION

Underground mines pose many hazards to workers. One of the major hazards is rock fall. To minimize this hazard and to stabilize the mine to maximize worker safety, sprayable concrete, known as shotcrete, is applied to the walls and ceiling of mine drifts, i.e., tunnels. In uranium mines, in addition to stabilizing the drift, shotcrete is used to reduce worker exposure to hazardous radiation.

Traditionally, the application of shotcrete has been done with a heavy machine equipped with an articulated boom and spray nozzle. This system is controlled by an operator who stands near the equipment and controls the system via an input device. This is a harsh and dangerous job; operators are required to wear respirator masks due to the amount of particulates produced. It would be ideal if the control of the system could be automated, thus removing workers from the hazardous environment. Using an autonomous system eliminates the need for operators to build up the necessary training and experience to achieve proficiency and consistency in shotcrete application techniques.

Though there are many robotic shotcrete systems, few are capable of autonomous shotcrete application. Of those able to perform shotcrete application autonomously, none are able to generate spray trajectories for entirely unknown surfaces. The autonomous shotcrete robot discussed by Girmscheid and Moser in [1, 2] is able to apply shotcrete on smooth blasted excavations and profiles drilled by tunnel boring machines but it is not designed to apply shotcrete to unknown irregular surfaces. Similarly, the work in [3] makes use of the design profile of a tunnel to generate shotcrete trajectories, an approach that is also unable to adapt to irregular surface geometries. The authors of [4] discuss the feasibility of two possible approaches to trajectory generation, using LiDAR (Light Detection and Ranging) scans or ultrasonic sensors to position the end-effector, but have not published any work where they generate spray trajectories. To the best of the authors' knowledge, there are no autonomous shotcreting robots available from manufacturers nor is there any published work in which shotcrete robots are able to autonomously generate spray trajectories in an unknown environment without any operator input.

For the purposes of this work, the articulated boom on a shotcrete system can be thought of as a robotic manipulator and it will be referred to as such in this paper. For the shotcreting task, the manipulator needs to position itself orthogonal to the mine surface, execute a motion that maintains a certain distance from the mine surface, and cover a specified region that may require the robot to change locations in order to complete the shotcreting process. Figure 1 shows the trajectory that would be required to shotcrete a flat square.

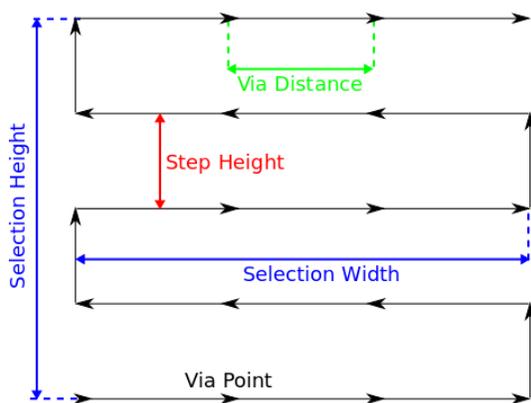


Fig. 1. Trajectory Generation Template



Fig. 2. Trajectory Overlaid on Mock-Up Mine Surface (Via Points Used for Angle Shown in Green, Distance in Blue, and Missing Points in Red)

A shotcrete trajectory generation algorithm requires some parameters to be set for the specific application,

while others remain universally constant. Specifically for shotcrete, the sprayer distance from the surface and angle relative to it are of the highest importance. Optimal distance and orientation maximizes the amount of adhesion and minimizes rebound [5]. Orientation requirements dictate that the nozzle must always be normal to the surface to minimize rebound. Distance requirements are dependant on factors such as nozzle shape, shotcrete composition, pressure, and flow rate. When the sprayer is too far away from the surface, not enough shotcrete sticks to the surface and when the sprayer is too close, shotcrete will bounce off the surface. In both cases, either shotcrete is wasted or the applied layer is of insufficient thickness. The former is uneconomical and the latter is unsafe. The step height shown in Figure 1 is determined by the desired offset and the amount of spread from the shotcrete spray nozzle. Parameters like nozzle shape, shotcrete composition, pressure, and flow rate all affect the spread of the shotcrete spray which dictates the necessary nozzle offset and step height.

Via points allow the robot to follow the uneven surface of the mine. In the example shown in Figure 1, there is no need for via points at locations along the horizontal lines, but when the image is overlaid on the mine surface as shown in Figure 2, the need for via points becomes apparent. Certain sections of the surface protrude out, meaning the spray nozzle must move back at that point to maintain optimal distance. The red circles highlight where via points must be added to compute and achieve optimal distance and the blue circles show existing via points that must maintain optimal distance. The surface normals must be calculated at each via point as the surface is not a single plane. The green circles highlight via points that require a significantly different angle to maintain perpendicularity at that point on the mine surface. Figure 3 shows the importance of choosing an appropriate radius in which to calculate surface normals. If too small of a radius is chosen, some surface contours may be missed. In practice small obtrusions are best ignored, this can be accomplished by choosing a larger radius for calculating surface normals. Since the intention of this algorithm is to be useful at any scale, the surface normal radius can be changed by the user but the default values are chosen for a typical underground uranium mine. The via distance is the amount of space between via points and can also be modified by the user. Determining the via distance requires knowledge of how rough the surface is likely to be. Via points can be calculated at every data point measured by the scanner, however, a high via point density leads to longer computation time and can become detrimental to the accuracy of the shotcrete application. Figure 4 shows how choosing too high a via point density leads to a convoluted spray path that may be less effective than a lower via point density. In the diagram it would have been sufficient to use the wider via point spacing. When the finer spacing was used the additional via points caused the sprayer to move along a trajectory that is unnecessarily complex. In practice, small obtrusions on the mine surface are better ignored than attempting to orient the sprayer orthogonally to the surface at every point.

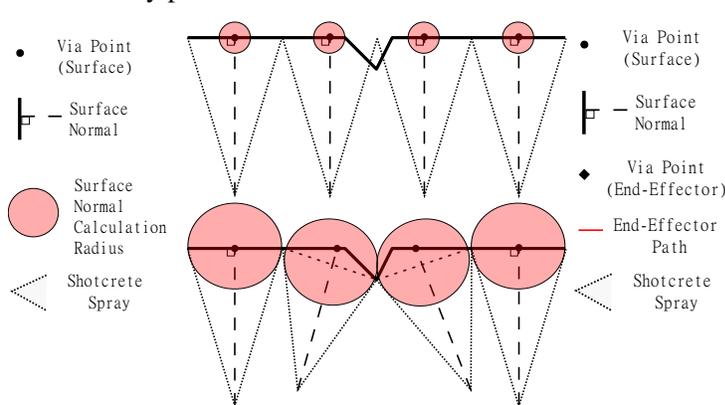


Fig. 3. Effect of Surface Normal Radius Size

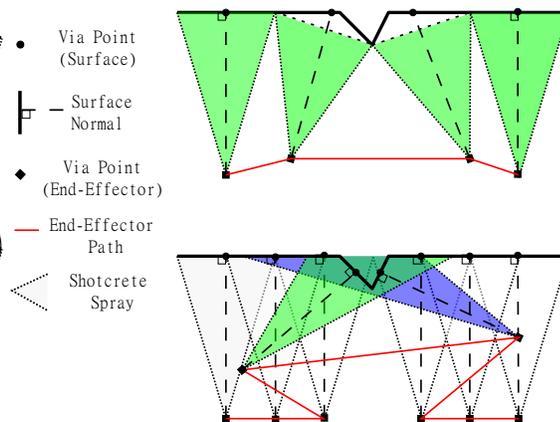


Fig. 4. Effect of Via Point Spacing



Fig. 5. Prototype System and Mock-up Mine

## 2. PROTOTYPE SYSTEM AND MINE MOCK-UP

In order to develop the trajectory generation algorithm for the automatic shotcreting application, a prototype system was developed to represent the shotcrete equipment. The prototype is comprised of a Clearpath Robotics Husky Unmanned Ground Vehicle (UGV) equipped with a Denso VP Series 6-axis articulated robot manipulator and a SICK LMS-150 LiDAR scanner mounted on a nodding head. The nodding head allows the 2D LiDAR to generate 3D point clouds. A custom built trailer was attached to the Husky to hold the Denso controller. The complete set-up is shown in Figure 5.

The software for the system was developed using the Robot Operating System (ROS - <http://www.ros.org> [6]) and the Point Cloud Library (PCL - <http://pointclouds.org>).

In order to replicate a mine environment in a controlled laboratory space, a 1:3 scaled mock-up of a mine was built with similar surface features as those found in a real mine. The mock-up mine was constructed of a wooden structure covered with foam sheets and cut-out foam blocks to represent rock features as shown in Figures 2 and 5. The mock-up mine represents two walls and a portion of the ceiling for a typical drift. Although there are many flat sections in the mock mine, the sensor noise from the LiDAR produced a point cloud that had a surface roughness comparable to the surfaces found in underground uranium mines.

## 3. TECHNIQUES FOR GENERATING TRAJECTORIES

Generating the template trajectory shown in Figure 1 on an uneven surface such as those found in a mine drift is not a simple task. In [1] the authors offered two approaches to generating shotcrete trajectories. Their first approach was termed semi-autonomous and relies on the user defining the trajectory while the robot executes it autonomously. The fully automated approach does not take into account the surface it is applying shotcrete to but simply moves along a pre-programmed path similar to the one shown in Figure 1. In reality, the pattern is to be generated on a surface that may contain sections in front, above, beside, or angled relative to the robot. Shotcrete is most effectively applied when the spray nozzle is normal to the surface to minimize rebound and maximize adhesion [5]. The research in [3] positions the nozzle normal to the surface, however, it relies on the design profile of the tunnel to generate spray trajectories and normal vectors rather than using a 3D scan of the mine environment. The authors of [7] use LiDAR scans to position the spray nozzle but not to generate spray trajectories.

The work presented in this paper does not only generate trajectories in which the spray nozzle remains nor-

mal to the surface, but also generates trajectories that follow the contour of the mine surface without requiring prior knowledge of its shape. As well, if the shotcrete area is larger than the manipulator's workspace, the robot will autonomously re-position itself.

### 3.1. Flattening Approach

One attractive solution to generating a 2D trajectory on a 3D surface is to convert the 3D surface to a 2D surface through a flattening process. Earth has been represented on flat surfaces many times throughout history, but all projections distort the original in some way. Most often, either the shapes are distorted but the scale remains the same, or the shape is maintained and the scales distorted. A summary of flattening approaches can be found in [8], in which the authors divide the approaches into two categories: geometric and mechanical methods. The classic geometry flattening method is called the "Triangular Patches Flattening Method" where the Non-Uniform Rational B-Spline (NURBS) surface is represented using a mesh of quadrilaterals that can each be subdivided into two triangles. The corners of the quadrilaterals are called nodes and the nodes are mapped to the 2D surface using the principal of increasing inner angle. The mechanical methods treat the connection between each node as a spring and try to minimize the energy of the surface as it is flattened, thus reducing the amount of distortion during the flattening process. NURBS surfaces are not always flattenable, so the "Flattenable Laplacian Mesh" is presented in [9] as an alternative mesh representation and the local flattenable perturbation approach is discussed explaining how such a mesh is flattened.

Flattening approaches are often used when clothing or sheet metal objects are manufactured from cut pieces of flat material. The "Woven Cloth" approach is well discussed in [10, 11], using a mechanical flattening method where the cloth's weft (horizontal) and warp (vertical) fibres are considered in-extensible but the angle between them flexible. Further energy based approaches are explored in [12] where popular methods are compared to their angle and area energy based technique. Woven cloth techniques often produce discontinuities like cuts or overlapping folds, which may be suitable for clothing design but causes unpredictable shotcrete spray trajectories.

Though there are many approaches to flattening surfaces, nearly all result in some form of distortion. The distortion can cause the spray trajectories to leave unsatisfactory shotcrete thickness, either too much or too little. Specific feature curves can be chosen such that they are not distorted, as presented in [13, 14], however, that would require the same amount of operator instruction as manually selecting the shotcrete spray path.

### 3.2. Path Planning Approach

An ant crawling along the mine surface would experience it as a 2D plane. Though the ant is moving in three dimensions, it only has two independent directions of travel. When trajectory generation is approached in this way, the problem becomes one of path planning. To generate a trajectory like that in Figure 1 the ant simply needs to walk the selection width, make a 90° turn, walk the step height, and make another 90° turn. The ant can then alternate directions and repeat until it has covered the entire selection area. The two challenges with this approach is travelling in a straight line across the surface and turning 90°.

For testing purposes, the algorithm assumes the selection area begins on a vertical section. While this is a reasonable assumption to make, had this technique been implemented in the final trajectory generation algorithm the surface normal at the start of the selection area would be used to determine an appropriate direction for the trajectory to begin with. Using Figure 6 as an example selection, the first via point ( $\mathbf{P}_1$ ) is chosen as the bottom corner of the selection. The second via point ( $\mathbf{P}_2$ ) is simply constrained to be within the same vertical height as the first via point and the via distance away. The third and all subsequent via points ( $\mathbf{P}_3, \mathbf{P}_4$ ) are calculated by moving the via distance along the vector created by the previous two via points ( $\mathbf{V}_{12} = \mathbf{P}_1\mathbf{P}_2$  &  $\mathbf{V}_{23} = \mathbf{P}_2\mathbf{P}_3$ ). The actual via point is found using PCL's "nearest neighbour" search in the

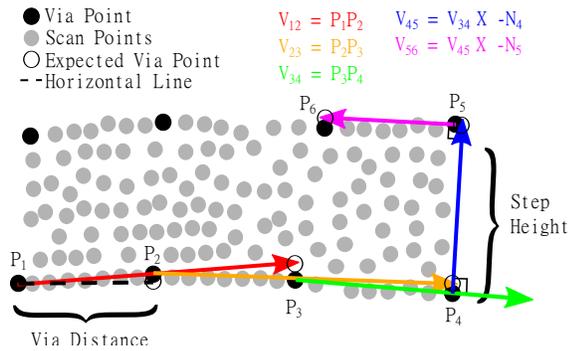


Fig. 6. Determining Via Points Using Path Planning

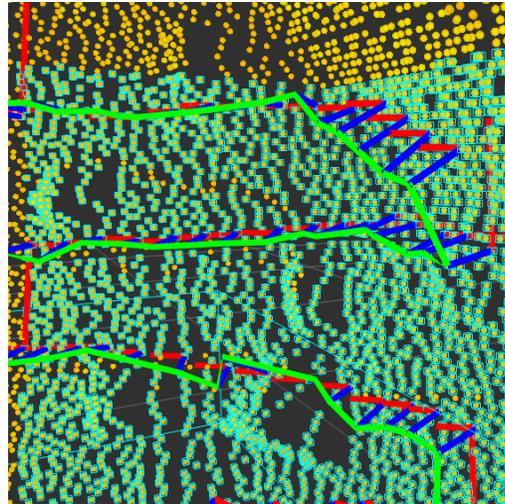


Fig. 7. Trajectory Generated Using Path Planning (Direction Vectors in Red, Surface Normals in Blue, and End-Effector Path in Green)

area of the expected via point. At the edge of the selection area the  $90^\circ$  turn must be made. The direction vector to the expected point ( $\mathbf{V}_{45}$ ) is calculated using the cross product of the vector from the previous point ( $\mathbf{V}_{34}$ ) with the negative vector of the surface normal at the current point ( $\mathbf{N}_4$ ). The next expected via point location is then determined using that direction vector ( $\mathbf{V}_{45} = \mathbf{V}_{34} \times -\mathbf{N}_4$ ) and the step height. The same steps are done for the next  $90^\circ$  turn ( $\mathbf{V}_{56} = \mathbf{V}_{45} \times -\mathbf{N}_5$ ), then the path continues horizontally until reaching the starting edge of the selection area. The arrows in Figure 6 show the direction vectors passing through their two points and ending at the expected via point location. Since the point cloud is not a continuous surface there is no guarantee there will be a data point at the expected location, but with sufficient scan resolution there will be a measured point reasonably close.

Since the robot is able to detect areas of insufficient shotcrete thickness, the path planning method is viable, however, in practice it did not consistently yield a satisfactory trajectory. An example trajectory using this method is shown in Figure 7. Though the path does not fail entirely, the problems become evident - via points are not sufficiently constrained to vertical and horizontal paths. The direction vectors proved unreliable, so to test the algorithm in a best-case scenario, the selected area was chosen to be mostly vertical and the direction vectors were replaced with horizontal vectors. The direction vectors shown in red, surface normals in blue, and end-effector path in green show how the “nearest neighbour” search and expected via points approach causes the spray trajectory to become uneven.

### 3.3. Plane Intersections

One simple technique to generate a shotcrete trajectory is to calculate the intersection between a horizontal plane and the surface. Though the robot may not be on a level surface, it can use its built in Inertial Measurement Unit (IMU) to determine its orientation with respect to gravity and determine a level plane. By using the “crop box” function of the PCL library the software can inflate the plane to a rectangular prism of a desired thickness and crop all surface data points that do not lie within the box. Using the “crop box” approach reduces computation time since it eliminates the need to form a mesh of the mine surface and calculate the intersection between the mesh and a plane. Moving the crop box by the user specified step height allows the algorithm to generate a series of lines across the detected surface. These lines can then be sorted and downsampled to produce the via points necessary for a trajectory. If the surfaces to be scanned or

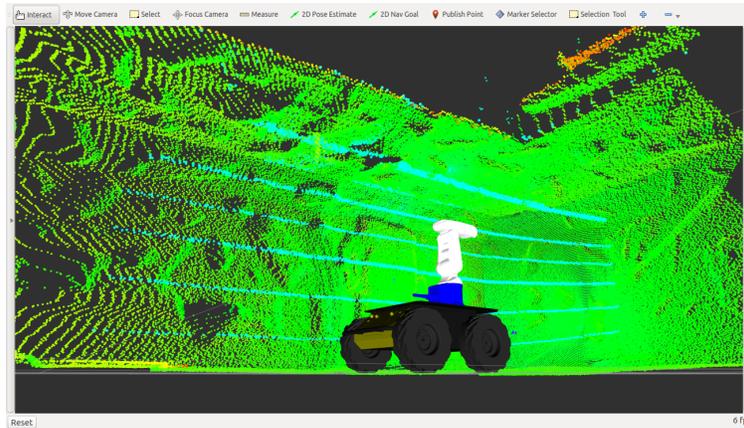


Fig. 8. Horizontal Plane Intersections

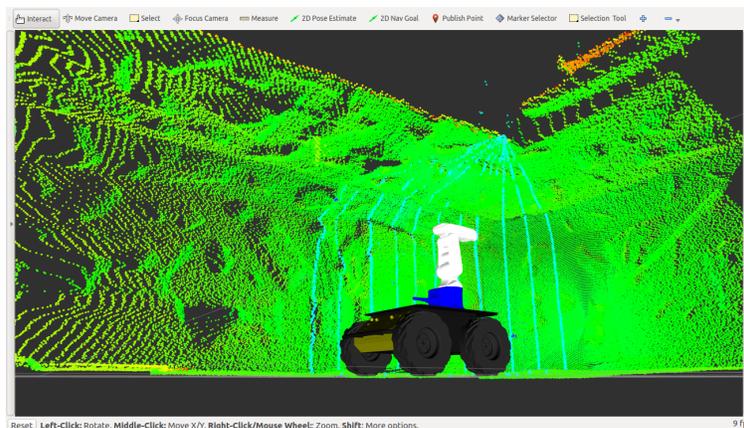


Fig. 9. Yaw Plane Intersections

sprayed were all vertical this technique would be simple and effective, however, when applying it to angled or horizontal sections the algorithm may fail. The top plane intersection in Figure 8 shows how using a rectangular prism and point cloud intersection on a nearly horizontal surface yields a line width of too many points. As well, the step height measured along the surface will not be correct, since the step height was measured along the vertical axis of the robot.

The next technique investigated was keeping the plane centred on the robot and rotating around the robot's axes as shown in Figures 10 and 9. Once again, using a fixed step angle will not yield a consistent step height. Moreover, the lines will converge at a point on a non-orthogonal wall. Consistent step height can be achieved by sorting the points on a line and determining the distance along the surface, then calculating the angle from the robot centre to the point at the appropriate step height. Using that angle the plane can be rotated by the appropriate amount to yield consistent and accurate step heights, but the converging patterns cannot be avoided. If the crop plane is centred on a point on the mine surface and rotated to an angle normal to the surface, some converging lines disappear but the overall trajectory is still insufficient as shown in Figure 11.

The proposed algorithm implements a combination of the approaches discussed.

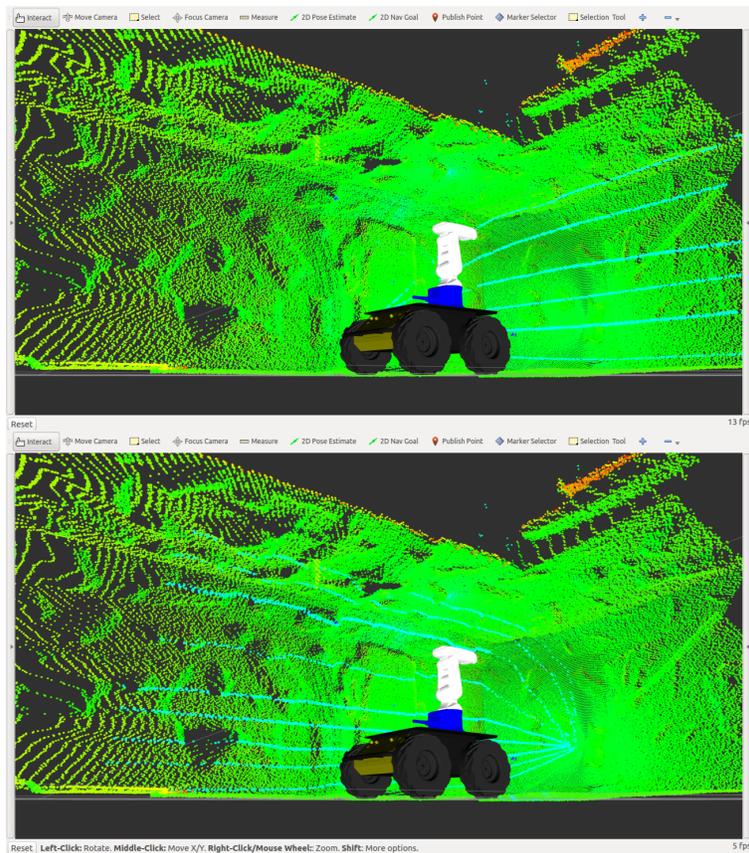


Fig. 10. Pitch and Roll Plane Intersections

#### 4. ALGORITHM USED FOR TRAJECTORY GENERATION

The final algorithm chosen for implementation uses a combination of the yaw plane intersections from Figure 9 for the wall sections and the pitch/roll plane intersections from Figure 10 for the ceiling sections. To generate trajectories the algorithm uses the following structure:

1. Compute horizontal plane intersection and sort line (in this work, a sorted line is a line of points which are sorted by distance measured along the surface from which they lie)
2. Compute vertical plane intersection and sort line
3. Use horizontal line to generate vertical ribs with sorted points
4. Use vertical line to generate horizontal ribs with sorted points
5. Select first point of each vertical rib, calculate surface normal, apply offset, and store trajectory point
6. Select a point “step height” away from first point on each rib (in reverse order), calculate surface normal, apply offset, and store trajectory point
7. Repeat Step 6 until trajectory points have been generated for all vertical ribs, alternating direction
8. Repeat Steps 5-7 with horizontal ribs

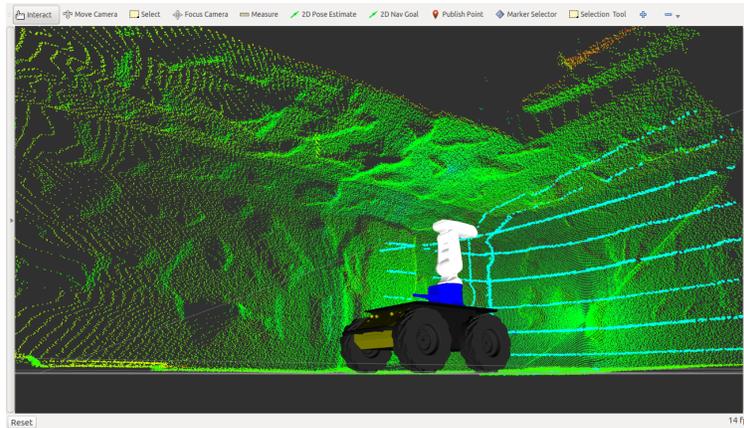


Fig. 11. Normal-to-Surface Plane Intersections

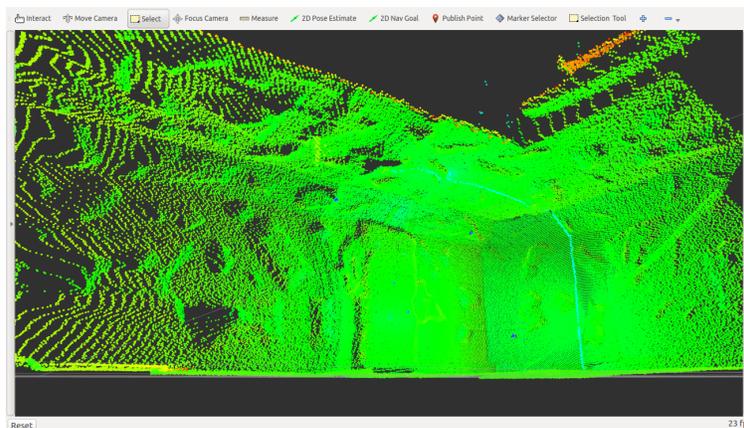


Fig. 12. Vertical Line from Plane Intersection

9. Drive forward if selected area is larger than manipulator's workspace and repeat from Step 1

Figure 13 shows the final trajectory. The rib location points calculated in steps 1-2 can be seen as two four point lines in 14. The sorted, downsampled ribs generated in steps 3-4 are shown in Figure 15, and the surface path in Figure 16. It should be noted the spray head will be instructed to shut off while it is moving from the final trajectory point of the wall section to the initial trajectory point of the ceiling section.

#### 4.1. Extracting Plane Intersections

Steps 1-2 of the algorithm begin with selecting a vertical and horizontal plane intersection. A line of sorted points is produced from each plane intersection and points that are via distance apart are used as locations to calculate ribs. The locations where the ribs are to be calculated can be seen as two, four-point lines in Figures 14. The final point of the sorted vertical line is not the via distance from the previous point, but is included to ensure the trajectory covers the entire selected area.

#### 4.2. Sorting Vertical and Horizontal Ribs

At each point generated from the sorted vertical and horizontal plane intersections, a vertical or horizontal rib is generated as shown in Figures 15. The ribs, receiving their name due to the similarity in appearance to a human rib cage, have their points sorted using the line point sorting algorithm. Trajectories for the curved

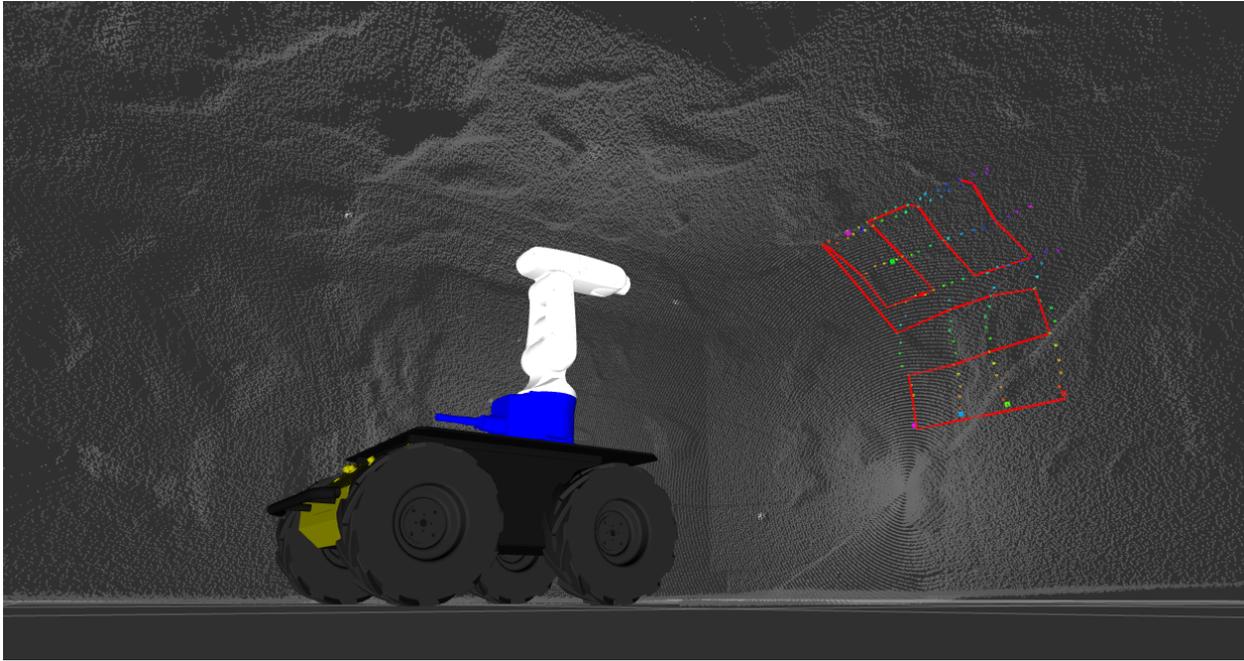


Fig. 13. Trajectory Generated Showing Surface Path

sections between wall and ceiling sections can be generated with either vertical or horizontal ribs, so exact determination of the transition point is not necessary.

#### 4.3. Selecting Trajectory Points

Once the ribs have been generated and sorted the trajectory points are selected to form the path shown in Figures 16. Beginning with the vertical ribs, the first point of the first rib is selected. The surface normal is calculated at that point and the trajectory point is offset from the surface along the surface normal by the amount defined in the configuration file. The orientation of the spray nozzle at each via point is determined as the negative value of the surface normal. Going through each rib in order, the first point of the rib is offset and added to the list of trajectory points. After the first point of each rib has been converted to a trajectory point, the same process is repeated using the ribs in reverse order. Rather than picking the first point on the rib, a point step height away from the previous point is used to calculate the trajectory point. This process is repeated while alternating the order of ribs from which the points are selected until the trajectory reaches the transition point from the wall section to the ceiling section. The process repeats itself on the ribs generated for the ceiling section to complete the trajectory generation as shown in Figure 16.

#### 4.4. Advancing the Robot

If the user selects an area to shotcrete that is outside the manipulator's workspace, it will autonomously navigate to an appropriate start position. Similarly, if the shotcrete area is larger than the manipulator's workspace, the robot base will move as necessary to ensure the entire selected area receives a shotcrete application. If placed in continuous operation mode by not selecting an area to shotcrete, the robot will autonomously apply shotcrete to an area the size of its workspace, then move through the mine applying shotcrete until the operator instructs it to halt.

The robot determines an appropriate position for shotcreting by orienting itself parallel to the mine surface at a predetermined offset to maximize the area of the mine surface that is within the manipulator's

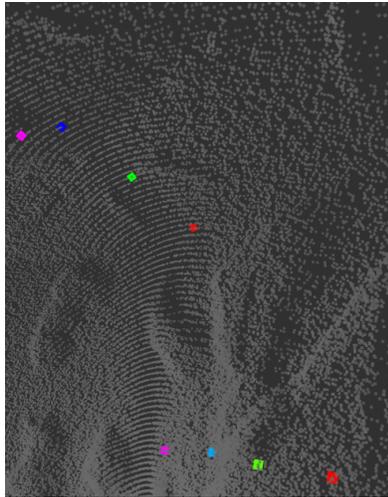


Fig. 14. Vertical and Horizontal Rib Location Points

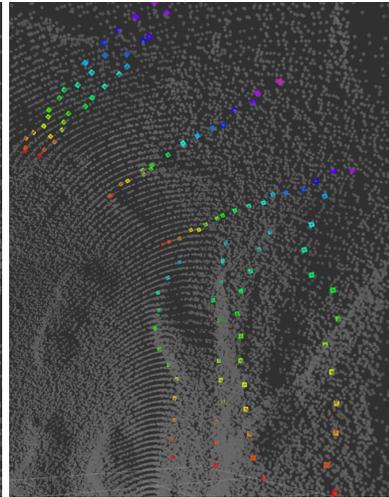


Fig. 15. Sorted and Downsampled Vertical and Horizontal Ribs

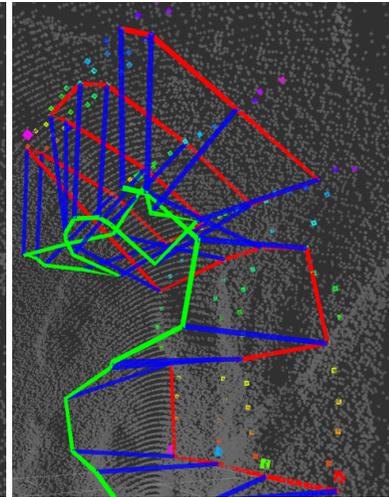


Fig. 16. Surface Path (Red), End-Effector Path (Green), and Offset Normals (Blue)

workspace. If the initial pose of the robot is beyond the manipulator's reach, it will search the selection area for the closest point and begin the shotcrete process after the robot has reached the appropriate location. If the robot begins at a location close enough to the mine surface to begin shotcreting, it will start immediately. When it completes the area within its workspace, it crops the completed section from the selection area leaving only the parts that have not received shotcrete. With only the incomplete section of the selected area remaining, the system selects the nearest point as a goal location and repeats the shotcrete application process until the entire selected area is complete.

When positioning itself, the robot determines a location one half its workspace beyond the nearest point to it, measured along the mine surface. By choosing a point further along the selected area it can minimize the number of times it has to move and maximize the amount of area it covers at each location. At the location it determines to be optimal, the surface normal is calculated and used to position the robot parallel to and offset from the mine surface.

Upon completion of the shotcrete task, the system is able to accurately estimate the thickness of the applied shotcrete using pre-shotcrete and post-shotcrete scans of the drift face as discussed in [15]. With the thickness information known, areas requiring further shotcrete application can be identified.

## 5. CONCLUSIONS

An autonomous path generation algorithm for automating the shotcreting process in underground mines has been developed. Using 3D point cloud data, the algorithm determines an appropriate trajectory for applying shotcrete to a drift's walls and ceiling. The system was designed to be robust, allowing its use in a variety of applications. Different applications require different configurations, so many parameters can be easily changed by the operator. Since this work was done with Cameco's underground uranium mine in mind, the default values have been selected to perform best there. Many other mines share similar properties, making the configuration parameters suitable for a wide variety of scenarios.

In the mock mine built for testing this robot, the generated trajectories were satisfactory for wall, ceiling, and combined sections of the mine. All of the figures shown were generated using actual data from testing the system in the mock mine, and the generated trajectories were executed successfully.

Alongside this work autonomous shotcrete thickness estimation was performed. This makes the application process much more reliable and robust, since it is able to determine areas of insufficient thickness and apply shotcrete to correct it. Next steps will be verifying the application process using real shotcrete applied in an actual mine.

## ACKNOWLEDGEMENTS

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and Cameco Corporation. Dr. Steve Asselin is thanked for helping with the French translation of the title and abstract for the paper.

## REFERENCES

1. Girmscheid, G. and Moser, S. "Fully Automated Shotcrete Robot for Rock Support." *Computer-Aided Civil and Infrastructure Engineering*, Vol. 16, No. 3, pp. 200–215, 2001.
2. Moser, S. and Girmscheid, G. "Fully Automated Shotcrete Robot." In "Proceedings of the 16th International Symposium on Automation and Robotics in Construction," pp. 385–390, 1999.
3. Nabulsi, S., Rodriguez, A. and Rio, O. "Robotic Machine for High-Quality Shotcreting Process." In "Proceedings of the 41st International Symposium on Robotics and the 6th German Conference on Robotics," pp. 1137–1144, June 2010.
4. Tibbs, M., Tyler, O., van Duin, S. and Baafi, E. "Autonomous Thin Spray-On Liner Application in Irregular Tunnel and Mine Roadway Surfaces." In "Proceedings of the 30th International Symposium on Automation and Robotics in Construction," pp. 1456–1463, 2013.
5. Ginouse, N. and Jolin, M. "Investigation of Spray Pattern in Shotcrete Applications." *Construction and Building Materials*, Vol. 93, pp. 966–972, 2015.
6. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R. and Ng, A. "ROS: an Open-Source Robot Operating System." In "Proceedings of the ICRA Workshop Open Source Software," , 2009.
7. Rodriguez, A.J. and Rio, O. "Analysis of Real Time Technical Data Obtained While Shotcreting: An Approach Towards Automation." In "Proceedings of the ECCOMAS Thematic Conference on Computational Methods in Tunnelling," , 2007.
8. Li, S. "An Overview of Flattening Methods for Complex Surface." *Journal of Information and Computational Science*, Vol. 11, No. 1, pp. 323–333, 2014.
9. Wang, C.C.L. "Towards Flattenable Mesh Surfaces." *CAD Computer Aided Design*, Vol. 40, No. 1, pp. 109–122, 2008.
10. Aono, M., Breen, D.E. and Wozny, M.J. "Fitting a Woven-Cloth Model to a Curved Surface: Mapping Algorithms." *Computer-Aided Design*, Vol. 26, No. 4, pp. 278–292, 1994.
11. Wang, C.C., Tang, K. and Yeung, B.M. "Freeform Surface Flattening Based on Fitting a Woven Mesh Model." *CAD Computer Aided Design*, Vol. 37, No. 8, pp. 799–814, 2005.
12. Xue, J., Li, Q., Liu, C. and Xu, M. "Parameterization of Triangulated Surface Meshes Based on Constraints of Distortion Energy Optimization." *Journal of Visual Languages and Computing*, Vol. 46, pp. 53–62, 2018.
13. Wang, C.C. "WireWarping: A Fast Surface Flattening Approach with Length-Preserved Feature Curves." *CAD Computer Aided Design*, Vol. 40, No. 3, pp. 381–395, 2008.
14. Zhang, Y. and Wang, C.C.L. "WireWarping++: Robust and Flexible Surface Flattening with Length Control." *IEEE Transactions on Automation Science and Engineering*, Vol. 8, No. 1, pp. 205–215, 2011.
15. Wrock, M.R. and Nokleby, S.B. "Robotic Shotcrete Thickness Estimation Using Fiducial Registration." In "Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference," pp. 1–11, 2018.