

Towards Topological Exploration of Abandoned Mines

Aaron Morris, David Silver, Dave Ferguson, and Scott Thayer

Robotics Institute

Carnegie Mellon University

Pittsburgh, PA, USA

{acmorr, dsilver, dif, sthayer}@cmu.edu

Abstract—The need for reliable maps of subterranean spaces too hazardous for humans to occupy has motivated the use of robotic technology as mapping tools. As such, we present a systemic approach to autonomous topological exploration of a mine environment to facilitate the process of mapping. This approach focuses upon the interaction of three high-level processes: topological planning, intersection identification and local navigation. Topological planning tasks the robot to investigate stretches of mine corridor for the purpose of collecting data. Intersection identification converts sensory input into topological components used to construct an online topological map and provide the robot with a global sense of position. Local navigation transforms topological exploration objectives into robot actuation enabling traversal of mine corridors. These processes are described in detail with results presented from experiments conducted at a research coal mine near Pittsburgh, PA.

I. INTRODUCTION

As discussed in previous mine mapping work [1], [2], [3], [4], abandoned mines pose a significant threat to both surface life and nearby subterranean operations. Mine maps, the primary source of information on mine structures, are key components in resolving this threat and preventing mine-related disasters; however, in the case of abandoned mines, such maps are often inaccurate and unreliable if they exist at all.

The components necessary to verify old maps or to produce new ones are (1) data on void structure and (2) a means of acquiring this data. To date, direct observation from within the void has been the most reliable method for obtaining quality data. The caveat, however, to establishing a physical presence within the mine is that human contact is often precluded due to the hostile nature of mine environments, especially when upkeep has been abandoned. As a result, robotic technologies have been applied to the domain of abandoned coal mines to offer nonhuman methods of observation and data collection. [2], [3], [5].

While robots can physically endure the harsh conditions of abandoned mines, mobility and autonomy for a robotic system is a significant challenge. Mine floors, often covered with fallen timbers, roof bolts, collapsed rock, and countless other mine-related artifacts, exhibit rugged terrain that is difficult to negotiate. Mine walls and ceilings also possess projecting obstacles that challenge robot maneuverability. In addition, the room-and-pillar mine structure prevalent in abandoned coal mines is problematic to exploration; the combination of a highly cyclic design with



Fig. 1. Groundhog, our current robotic mine mapping platform

no external communication (including GPS) makes it easy for a robot to become lost. Therefore, along with physical robustness, an autonomous mine mapping robot must be equipped with a highly reliable navigational framework.

Our current research is aimed at addressing each of the aforementioned challenges. We are interested in both the creation of systems that can robustly explore mine environments and the development of algorithms that can generate accurate maps of these environments. Previous work in this area [3] has alleviated some of the major exploration challenges by dealing primarily with the task of portal inspection. Portal inspection restricts exploration to the primary entrances of an abandoned mine and confines the robot into traversing a single corridor. This simplifies navigation and eliminates issues concerning global localization and loop closure. The push for more intricate mine inspection tasks, however, demands that these issues be resolved. To enable functionality such as perimeter discovery or full exploration, more capable navigation and global positioning are required if the robot is to successfully explore multiple corridors.

In this paper, we present a systemic approach to robotic exploration of a room-and-pillar [6] mine. This approach is based upon the interaction of three core concepts: topological planning, intersection detection, and corridor navigation (Figure 2). Topological planning maintains a topological representation of the mine and generates navigation

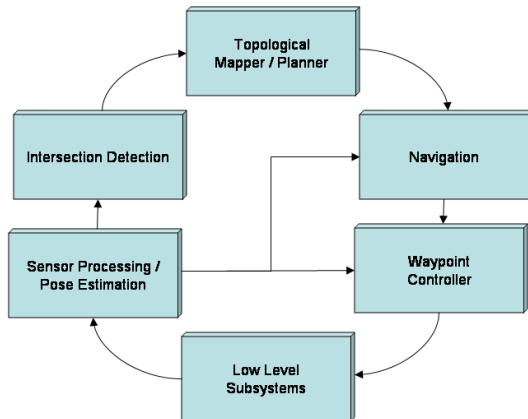


Fig. 2. System overview.

objectives using this topological framework. Intersection detection monitors sensory input and identifies corridor intersections, which create the nodes of the topological representation. Local navigation decomposes navigation objectives into motion planning goals that direct the robot along corridors for the exploration of new areas.

The remainder of this paper discusses the concepts of our topological exploration method. In Section II we discuss related research in this area. We then present our system in Section III and results from an experimental mine in Section IV. We conclude with discussion and extensions.

II. MOTIVATION FOR TOPOLOGICAL EXPLORATION

Towards achieving our mine mapping goal, we have constructed a 700 kg custom-built ATV-type robotic platform known as **Groundhog** (Figure 1) that is physically tailored for operation in the harsh conditions of abandoned mines [3]. Groundhog has been used extensively in both test and abandoned mine environments, accruing hundreds of hours of mine navigation with 8 successful portal entry experiments in the abandoned Mathies mine outside of Pittsburgh, PA. From these experiments, log data has generated globally consistent large-scale maps (Figure 3) using offline techniques [1], [2]. Over the course of its lifetime, Groundhog has evolved into a system that is highly proficient at autonomously traversing and mapping isolated mine corridors, successfully navigating over 2 km of abandoned mine.

For the next phase of Groundhog’s development, we wish to extend exploration tasks beyond isolated corridors and towards a complete cyclic mine structure. In this paper, we have taken the first step towards full exploration by performing acyclic or tree exploration. The challenges involved in cyclic and acyclic exploration are nearly identical with the exception of loop closure. Loop closure is still a difficult problem in globally consistent mapping research and one we are currently investigating separately. Instead, this paper addresses the pertinent requirements that take

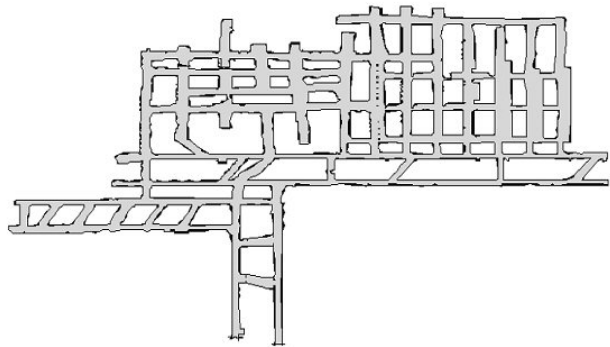


Fig. 3. This map was generated from data acquired during experimentation and utilizes our offline globally consistent mapping techniques. It shows the highly cyclic nature of room-and-pillar mines.

Groundhog from portal inspection to multi-corridor exploration. These requirements include (1) a representation of the mine for global localization, (2) a task-level planner that can decompose an exploration objective into intermediate navigation goals, and (3) a navigation system that can realize the navigation goals through path planning.

Topological representations [7] coincide nicely with the inherent structure of room-and-pillar mines. Room-and-pillar mines consist almost exclusively of narrow corridors and corridor intersections (see Figure 3). A topological map is a graph representation of a given environment where the nodes of the graph correspond to distinct locations in the environment and the edges correspond to a direct path between two such locations. For mines, nodes and edges correspond to intersections and corridors, respectively. In addition, topological representations exhibit several advantages over grid or feature-based approaches. Such advantages include a compact data representation and a reduction in processing time for the construction, manipulation, and utilization of these maps [8].

Implicit in representing a mine as a topological map is that the robot has some way of determining node locations from its sensor data, as well as the edges leaving each node. Once on an edge, the robot must be able to follow that edge to the next node. A topology that is detectable from local sensor data is said to be embedded in the robot’s environment. Embedded topologies exhibit useful properties in the context of task-level planning for exploration.

For task-level planning, topological maps have proven to be very useful in robotic exploration tasks [9] [10]. Unexplored edges in a topological map correspond to unexplored regions of the robot’s environment. If the chosen topology is a roadmap [11], exploring every edge in a topological map will allow the robot to explore every part of the freespace.

If this roadmap should be a tree (a condition that is enforced in the context of this paper), then the environment

can be fully explored by maintaining a list of nodes that have unexplored edges. A sequence of edge traversals can then be planned [8] to direct the robot to traverse an unexplored edge. This process repeats until all edges are explored.

For navigation, three dimensional motion planning (2D position and 1D orientation) is necessary given that the robot must choose between different corridors at each intersection. To further complicate 3D motion planning, the robot must cope with the complex arrangement of obstacles inherent in these environments. As described in [3], obstacles can project from the ceiling and walls of mines, which places additional constraints on such a path planner.

Terrain analysis and select aspects of path generation were covered in previous work [1], [2]. The navigation system utilized in our topological framework builds upon the previous terrain analysis and path generation techniques by incorporating a nonholonomic 3D path planning algorithm adopted from work by Latombe [12]. This navigation routine, as well as all the other aforementioned system requirements, are described in the following section.

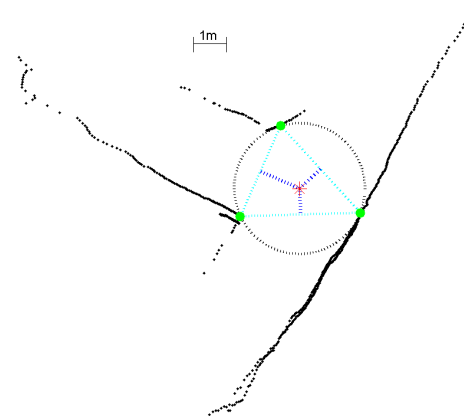
III. SUBTERRANEAN TOPOLOGICAL EXPLORATION

Our system for topological exploration in mine environments is based upon three core modules: a topological planner, an intersection detector, and a local navigator. These processes capture the topological representation, task-level planning, and navigation requirements discussed previously.

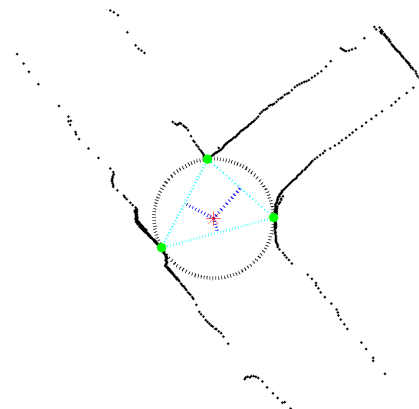
A. Topological Planning

At the task-level planning layer, the topological exploration module plans robot actions on an edge-by-edge basis. The mine is modeled as an acyclic graph where a fully explored mine will consist of N edges and $N+1$ nodes. Each edge corresponds to a corridor of the mine and each node represents either an intersection of corridors or the terminating point of a corridor, which we label as a terminal node.

The topological planner is instantiated with a root node and a trailing edge that is marked as unexplored. The edge is then traversed by the robot until either (A) an intersection is detected or (B) the passage becomes blocked. Under condition A, the previously traversed edge is marked as explored and the remaining M unexplored edges (where M is at least 2) are marked unexplored from a newly created node. The robot will then choose an unexplored edge to traverse and repeat the edge exploration process. Under condition B, the previously traversed edge is still labeled explored, but a terminal node is created. The robot will then begin backtracking along the previous edge until it resides at the previous node. From there it will explore the remaining unexplored edges of that node or proceed with backtracking. The exploration process terminates once the robot has arrived at the root node.



(a) Strong Node



(b) Weak Node

Fig. 4. Two examples of intersection detection. Node locations are shown in red, along with the associated Delaunay triangle, and tangents to the local Voronoi edges. In (a) the node is considered an intersection, because each tangent would lead the robot in a direction for which it can not see a boundary. The node in (b) is not considered an intersection, as one of the edges heads towards a clearly distinguishable boundary.

B. Intersection Detection

Our intersection detector is based on [13], in which we described a technique for extracting topological features from laser range scans that can be used to identify intersections. These features correspond to nodes of the generalized Voronoi diagram (GVD) [14]. The GVD is an embedded topology, and has been used in previous topological mapping and navigation systems [15] [16]. Features are extracted by computing the Delaunay triangulation [17] of a range scan, and keeping only triangles that have all three sides longer than a distance parameter α . These triangles correspond to Voronoi nodes with the property that each edge leaving the node is at least $\frac{\alpha}{2}$ away from every corner of the triangle. The interpretation of this property is that the Voronoi edge corresponding to the feature can be safely

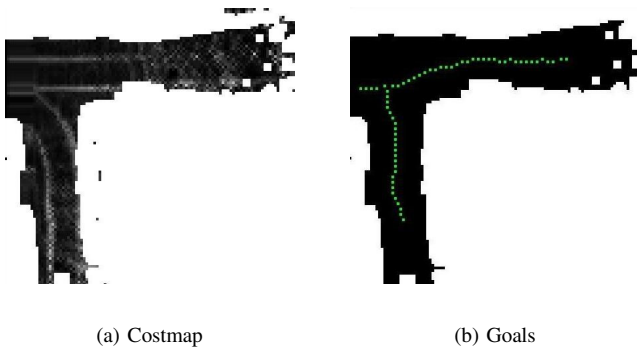


Fig. 5. Planning maps. (a) The 2.5D costmap and (b) potential goal positions.

traversed by a robot with a width less than α for at least a short distance.

While extracting GVD nodes from a single range scan provides candidate intersections, it is not sufficient for intersection detection. Sparse or noisy range data can often produce false positives in a single scan. This problem is solved by tracking features over multiple scans. For features from two different scans to match, all the properties of each feature (Voronoi node position and radius, and the position of the corners of the corresponding Delaunay triangle) must match within defined thresholds. For a feature to be considered as an actual GVD node it must be matched through a minimum number of consecutive scans. Experimental tuning of all pertinent parameters has yielded a reliable detection system.

After detecting a GVD node, one final step is necessary before adding a new node to the topological map. GVD nodes can be created by any sufficiently large concavity in the local structure. Such concavities occur often in the irregular walls of a mine corridor. The GVD nodes corresponding to these features are often termed "weak" nodes. Edges that leave a weak node towards a closed concavity are termed boundary edges. Eliminating these nodes and keeping only "strong" nodes results in a topology known as the reduced GVD (RGVD) [18]. The distinction between strong and weak nodes is dependent on the scale at which the local environment is perceived. For our purposes, the scale is defined in terms of the robot's size and sensor range, and so the distinction between strong and weak nodes is well defined.

RGVD nodes can be detected by checking to see if any edges leaving a candidate node are boundary edges. Such a test requires range data from a full 360 degrees around the candidate node. Given Groundhog's sensor configuration, this is not possible in one scan. Instead, two scans are combined from different vantage points, each along a different candidate edge. This requires Groundhog to actually drive through an intersection before it can add it as a node in the topological map. The local map formed

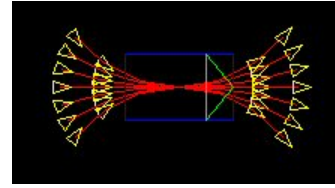


Fig. 6. The set of planning arcs used in path generation.

by these combined scans is also useful for node matching, as will be discussed in section V.

C. Local Navigation

Navigation is the intermediate process between topological exploration and vehicle actuation. The navigation module accepts topological objectives from the task-level planner, such as "turn and face corridor a " or "traverse corridor b ," and produces robot maneuvers. These maneuvers take the form of waypoints that the robot controller tracks and follows.

The navigation process utilizes the Sense-Plan-Act (SPA) architectural framework to transform exploration tasks into motor commands. In the navigational context, the SPA procedure accrues sensor data from the robot's environment (sense); creates an internal representation of the environment and plans a set of actions to achieve a navigation objective (plan); and executes this plan by maneuvering the robot towards a goal location (act). These SPA phases operate in a continuous cycle throughout the duration of exploration.

The sensing portion of SPA executes from a stationary position. Scanning lasers capture range measurements from the ceilings, floors, and walls in the robot's immediate field of view. This scan data, represented as a 3D Cartesian point cloud, is compressed into a 2D grid. The grid encodes the traversability of the terrain (also known as a 2.5D cost map) where each grid cell is marked either lethal or traversable conditioned upon the detection of an obstacle in that cell. Traversable cells are assigned with a local gradient assessment to estimate the cost of traversing that particular cell.

Navigational planning then utilizes this cost map in two phases:

- 1) Goal selection: the process of choosing a goal that coincides with the topological exploration objective
- 2) Path generation: the process of choosing a feasible set of motions that achieve the goal without colliding with obstacles

Goal selection decomposes the objective of the topological planner into pose goals within the robot's workspace. Since the robot must position and orient itself to align with unexplored corridors, goals and workspace are expressed in three dimensions (i.e. $\mathcal{R}^2 \times \mathcal{S}$). For each SPA cycle, the robot will choose a goal pose that positions the robot in (x, y) and orients the robot in θ .

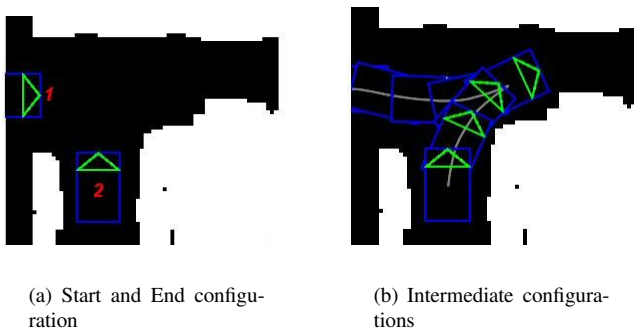


Fig. 7. An example of path generation

It is important to note that the greatest challenge in goal selection under our exploration framework is in picking goals that ensure continued progress for the next SPA cycle. A bad goal, one that places the robot facing a wall or an undesired corridor, can disorient the robot. In such cases, the robot can prematurely stop exploring a corridor. For this reason, goals are selected that coincide with the center of the corridor and face in the general direction of the corridor.

Path generation, the second phase of planning, creates paths in the aforementioned 3D workspace while considering the drive characteristics of the robot. The nonholonomic constraint imposed by the platform's Ackerman drive system makes this problem similar to the nonholonomic motion planning problem for car-like robots [12] [19]. For Groundhog, however, this problem is simplified in that Groundhog moves at a fixed and rather slow speed. As such, we can neglect acceleration and vehicle dynamics, and simplify the problem to a geometric search of paths through the workspace.

Our path generation algorithm is adapted from the techniques addressed in Latombe [20] and the MORPHIN algorithm [21]. A finite set of arcs (corresponding to a set of fixed steering angles) of uniform arc length are selected to represent the set of moves the robot can make from a given pose. Both forward and reverse arcs are included (corresponding to a positive or negative velocity, respectively) as well as straight lines (corresponding to a steering angle of zero). The arcs are explicitly stored as a finite collection of relative poses that describe the motion of the robot for each specified steering angle (Figure 6).

The workspace is discretized in both position and orientation to form a 3D state space used in the search routine. Position is divided into cells that match the cell size of the cost map and align with cell boundaries. The orientation is divided into k bearings between $[-\pi, \pi)$. The size of k can be varied to manipulate planning performance. A large k , for example, will generate shorter paths, but will also increase memory and computation time required to find a solution whereas a small k will generate faster solutions at

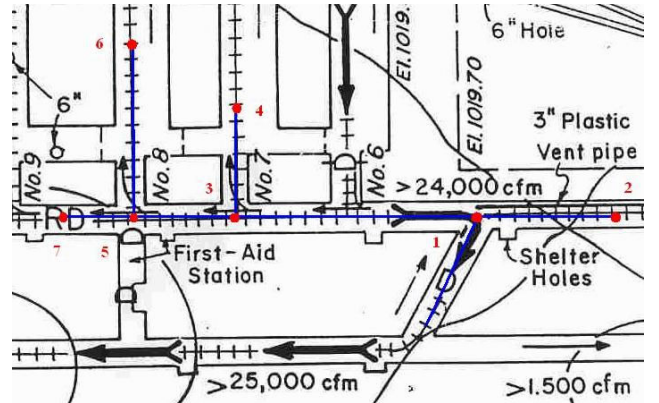


Fig. 8. A survey map of a section of the Bruceton mine, with node locations shown in red. The unterminated edge leaving Node 1 represents the path out of the mine.

the price of longer paths.

The search routine determines a collision-free path by performing a biased breadth-first search through the state space using the robot's arc set to generate successor states. The algorithm maintains a list of OPEN and CLOSED states. The OPEN list is initialized with the start state, which is mapped into the discrete state space $(x_0^d, y_0^d, \theta_0^d)$ from a continuous space start pose $(x_0^c, y_0^c, \theta_0^c)$. From the start pose, successor poses are calculated from the end poses for each arc in the arc collection. The c-space is dynamically determined by performing a collision check along robot's footprint at each terminating arc pose within the cost map. Only collision-free arcs will have terminating poses in $(x_i^c, y_i^c, \theta_i^c)$ mapped into a discrete states $(x_i^d, y_i^d, \theta_i^d)$ and added the OPEN list of states. The previously visited state is then placed on the CLOSED list and never revisited. The process continues to expand OPEN states until either a goal state is reached or no states reside on the OPEN list. If a solution is found, it is traced by following a set of back-pointers connecting transition states from the goal to the start. Otherwise, no solution is returned.

Actuation is the latter part of SPA that issues a chain of platform commands to drive Groundhog along the planned path solution. The solution is issued as a series of waypoints generated from the planned arcs. These waypoints are then tracked using feedback from Groundhog's pose estimator. Tracking waypoints rather than arcs prevents tracking errors from accumulating as rapidly.

IV. RESULTS

Experiments were conducted in the Bruceton research coal mine outside of Pittsburgh, PA. Each experiment consisted of starting Groundhog in a tree environment and proceeding with an autonomous exploration task. Over the course of each exploration, a topological map is constructed, along with a log of intersections detected. These experiments were designed to test the new capabilities of

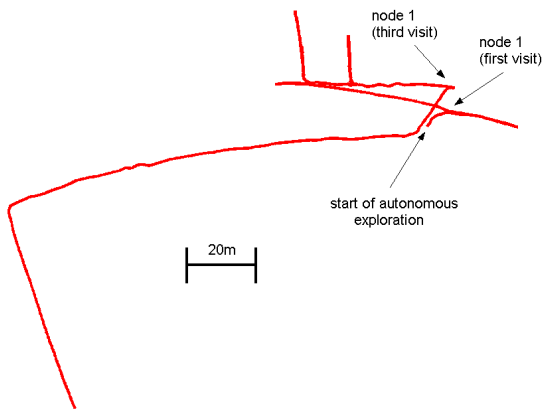


Fig. 9. Groundhog's path from online pose estimation during an experiment. Even though the pose estimates for a particular intersection differ each time it was detected, the topological map provides a mechanism for determining the correct location.

Groundhog, mainly a high level awareness of position, the capacity to choose and follow topological objectives, and the ability to navigate in a way meaningful to exploration (ie. turning corners).

Figure 8 shows the topological map built during the largest single experiment, overlaid on a survey map of the corresponding section of the mine. In this run, Groundhog was teleoperated from the portal to a position near the first intersection. It then proceeded to autonomously explore as much of the environment as was traversable, and then autonomously return to the portal. Over the course of 2 hours, Groundhog traversed more than 400m of mine corridor.

Figure 9 shows the path recorded from Groundhog's online pose estimator during this same experiment. While accurate enough in the short term, the online pose estimation exhibits considerable error over the long term, rendering it useless for high level navigation. However, by maintaining its position on the topological map, Groundhog was able to correctly determine when it had returned to an intersection it had already visited. It was thus able to fully-explore the mine environments used in our experiments and safely navigate its way back to the exit of the mine.

Over more than 20 hours of testing of the final system, and passing through more than 50 intersections, the intersection detector correctly identified 100 percent of relevant intersections. Furthermore, it never classified a weak node as a strong node. This sort of reliability is paramount to successful navigation in hazardous, abandoned mine environments. Figure 4 demonstrates the intersection detector in both the strong and weak case.

The local navigation module ran continuously throughout these experiments. During basic corridor traversal path planning took approximately 10 seconds per SPA cycle using Groundhog's 300 MHz processor. The previous

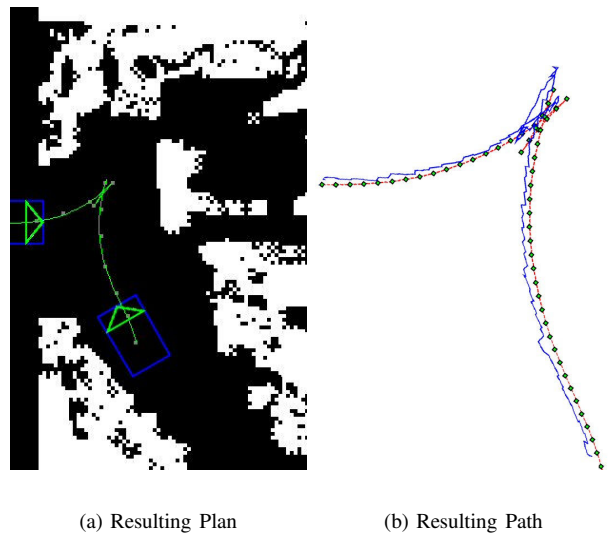


Fig. 10. An example of planning and waypoint following. The start and end configurations, as well as the planned path of 11 arcs, are shown in (a). The path followed by the robot according to its online pose estimation is shown in (b)

Groundhog navigation system required approximately 60 seconds under the same conditions. More complicated turning behaviors took between 30 and 60 seconds on average, depending on the complexity of the maneuver in question. An example path, along with Groundhog's tracking of said path, is shown in Figure 10.

V. CONCLUSION

In this paper, we have described the high level processes that enable our system to autonomously explore a complex mine environment. These processes consist of a topological planner, an intersection detector, and a local navigator. These processes, in conjunction with the existing Groundhog architecture, enable exploration of tree-structured mine environments. We have presented results from one of several experiments conducted in a mine environment that verify this approach.

This work has extended the capability of our robot to the point where it is able to perform a number of realistic tasks associated with mine safety and rescue. As well as mapping acyclic operational and abandoned mines, it is capable of exhibiting "go to" behavior, where it could be instructed to locate a particular intersection in a mine to check for structural damage or wounded personnel.

We are currently concentrating on extending our autonomy framework to handle cyclic mine environments. It is our belief that the topological properties of detected nodes can be combined with the local mine structure at these areas to allow for robust node recognition. Previous work [13] demonstrated the feasibility of this approach with 2D local maps. We are currently working to extend this approach to

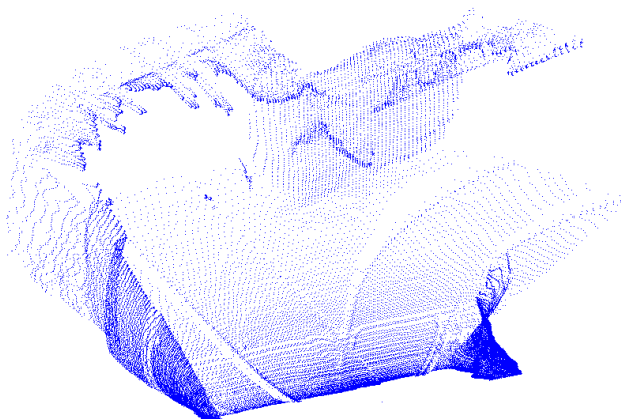


Fig. 11. A point cloud generated from a 3d scan at a corridor intersection in the Bruceston mine.

use detailed 3D information (Figure 11) as input to surface matching and recognition routines [22]. This will allow our system to better exploit the inherent uniqueness of irregular mine corridors and intersections to aid in loop closure. With such a capability, the exploration and navigation of general mine environments would be possible, completing the primary goal of the Mine Mapping Project.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Carnegie Mellon Subterranean Robotics and the Carnegie Mellon HSLAM groups for efforts directly related to systems development; the National Institute of Occupational Health and Safety, Mine Safety and Health Administration, Pennsylvania Department of Environmental Protection, and various people in the mining industry who provided guidance and support for this work; Workhorse Technologies, LLC for technical and operational support in all aspects of the project; and finally, Senator Arlen Specter for his continued support.

REFERENCES

- [1] S. Thrun, D. Hahnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker, "A system for volumetric robotic mapping of abandoned mines," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003.
- [2] D. Ferguson, A. Morris, D. Hahnel, C. Baker, Z. Omohundro, C. Reverte, S. Thayer, C. Whittaker, W. Whittaker, W. Burgard, and S. Thrun, "An Autonomous Robotic System for Mapping Abandoned Mines," in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*. MIT Press, 2003.
- [3] C. Baker, A. Morris, D. Ferguson, S. Thayer, C. Whittaker, Z. Omohundro, C. Reverte, W. Whittaker, D. Hahnel, and S. Thrun, "A Campaign in Autonomous Mine Mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, 2004.
- [4] A. C. Morris, D. Kurth, D. Huber, C. Whittaker, and S. Thayer, "Case studies of a borehole deployable robot for limestone mine profiling and mapping," in *International Conference on Field and Service Robotics (FSR)*, July 2003.

- [5] A. Nuchter, H. Surmann, K. Lingermann, J. Hertzberg, and S. Thrun, "6d slam with an application in autonomous mine mapping," in *IEEE International Conference on Robotics and Automation*, 2004.
- [6] A. B. Cummins and I. A. Given, Eds., *SME Mining Engineering Handbook*. Port City Press, 1973, vol. 1.
- [7] B. Kuipers and Y. Byan, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *J. Robot. Auton. Syst.*, vol. 8, pp. 47–63, 1991.
- [8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. MIT Press, 2001.
- [9] G. Dudek, M. Jenkin, E. Miliotis, and D. Wilkes, "Robotic exploration as graph construction," *Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 859–865, December 1991.
- [10] I. M. Rekleitis, V. Dujmović, and G. Dudek, "Efficient topological exploration," in *Proceedings of International Conference in Robotics and Automation*, Detroit, USA, May 1999, pp. 676–681.
- [11] J. Canny, "Constructing roadmaps of semi-algebraic sets I: Completeness," *Artif. Intell.*, vol. 37, pp. 203–222, 1988.
- [12] J.-P. L. (Editor), *Robot Motion Planning and Control*. Springer Verlag, 1998, vol. 229.
- [13] D. Silver, D. Ferguson, A. Morris, and S. Thayer, "Feature extraction for topological mine maps," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2004.
- [14] H. Choset and J. Burdick, "Sensor based planning, part II: Incremental construction of the generalized voronoi graph," in *Proc. of IEEE Conference on Robotics and Automation*. Nagoya, Japan: IEEE Press, May 1995, pp. 1643 – 1648.
- [15] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (slam): towards exact localization without explicit localization," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, pp. 125–137, Apr. 2001.
- [16] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset, "Hierarchical simultaneous localization and mapping," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, vol. 1, Oct. 2003, pp. 448–453.
- [17] F. P. Preparata and M. I. Shamos, *Computational Geometry*. Springer-Verlag, 1985.
- [18] K. Nagatani and H. Choset, "Toward robust sensor based exploration by constructing reduced generalized voronoi graph," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1999.
- [19] A. Kelly and B. Nagy, "Reactive nonholonomic trajectory generation via parametric optimal control," *International Journal of Robotics Research*, vol. 22, pp. 583–601, 2003.
- [20] J. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [21] R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, L. Katragadda, S. Koenig, G. Krishnaswamy, Y. Shinoda, W. R. L. Whittaker, and P. Klarer, "Experience with rover navigation for lunar-like terrains," in *Proceedings of the 1995 Conference on Intelligent Robots and Systems (IROS '95)*. IEEE, 1995, pp. 441 – 446.
- [22] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 1999.