Covering all the Bases: Robotic Surveying for Intra-Vehicular Robots in Microgravity

Ian D. Miller[1], Trey Smith[2], and Vijay Kumar[3]

***Abstract*— An important capability of mobile robots is the autonomous surveying and mapping of various environments. In particular, for Intra-Vehicular Robots (IVR), it is important to maintain regular situational awareness of various systems. There is therefore a need for a method to generate survey trajectories for the robot to regularly execute. These trajectories should allow for the mapping of a given area as thoroughly and rapidly as possible. In this work, we present a method for the generation of such trajectories for visual surveying using the Astrobee robots on the ISS. We finally show the efficacy of our method in the Astrobee simulator for surveying the JPM and Columbus modules.**

I. Introduction

Coverage Path Planning (CPP) is a well studied-problem in robotics [1]. It can be formulated in general as the problem of generating a path which covers a region in some sense, usually with the goal of doing so as rapidly as possible. A typical method for CPP in 2D is to decompose the region to be surveyed into cells which are easy to generate surveying primitives such as Boustrophedon "zig-zag" paths over [2]. These methods are simple, fast, and complete. However, they are typically intended for surveying of planar surfaces using ground robots, and IVR live in intrinsically 3-dimensional space.

These decomposition ideas have been extended to complex surfaces in 3 dimensions. Atkar et al. [3] compute a surface offset from the surface of interest for the robot to move in. They then detect so-called "critical points" by which to decompose this surface. For each decomposed section, they intersect the surface with a plane to compute a series of surveying loops. While this method would work for the more complex IVR geometry, the resulting trajectories are complex and nonlinear. In addition, execution of these trajectories may require lots of changes of robot orientation, which is not ideal for Astrobee.

Another class of CPP algorithms which are particularly powerful in high dimensional, complex environments where there may be motion constraints is sampling-based planning [4], [5]. These methods randomly sample points on the viewing manifold and maintain a tree with these points and the paths connecting them. They then search over this tree to find the final survey trajectory. These methods tend to be more computationally demanding, as well as generate final trajectories with less structure due to the random sampling.

[1,3]Ian D. Miller and Vijay Kumar are with the University of Pennsylvania GRASP Lab, Philadelphia, PA, USA. `{iandm, kumar}@seas.upenn.edu`

[2]Trey Smith is with the Intelligent Robotics Group at NASA Ames Research Center, Mountain View, CA `trey.smith@nasa.gov`

To address the drawbacks of these methods, we propose a new algorithm for surveying of IVR environments. In particular our algorithm seeks to generate a series of survey points and connect these points such that the robot's motion is in planes aligned to the major surfaces in the environment. Our contributions are as follows:

1) We describe our method for visual surveying of known geometric environments tailored specifically for IVR.
2) We release the code for our method publicly to aid in the advance of state-of-the-art in the field.
3) We demonstrate our method applied to the JPM and Columbus modules of the ISS in the Astrobee simulator.

II. Method

Our method, inspired by [2], [3] uses a hierarchical decomposition method. We assume that the approximate geometry to survey is known *a priori* and specified as the interior of a mesh $\mathcal{M}$. We begin by decomposing $\mathcal{M}$ into a set of orthographic charts $\mathcal{C} = \{C_i\}$. For each chart, we plan a Boustrophedon path in 2D following [2] and choose waypoints along these paths. We then project these waypoints back into 3D space and solve the Travelling Salesman Problem (TSP) to connect the individual charts. Finally, we refine these waypoints and prune as necessary in order to guarantee they are collision-free. Note that our method does not find trajectories for travelling between the waypoints as we assume the existence of a standard low-level planner to perform this task. We detail these steps in the following sections.

A. Chart Decomposition

Our primary sensor for surveying is a single RGB camera, however our method is equally appropriate for any projective sensor with a limited field-of-view. We would like to decompose the mesh in such a way that we minimize distortion from the perspective of this camera. A charts is a mapping covering a portion of the mesh and uniquely mapping every point on the submesh surface to a 2D coordinate. In general, multiple charts are needed to cover a mesh. These coordinates are commonly used for object texturing in computer graphics [6], which is effectively the end goal of our map procedure. We therefore use orthographic charts in order to map a 3 dimensional mesh coordinate onto a plane. As an additional benefit, an orthographic chart is defined by a normal vector which corresponds to the viewing angle over the chart.

In order to perform lhe chart decomposition, we begin by randomly picking a facet of the mesh and using its normal vector define a new chart. We then look at the neighboring facets in the mesh, project them to the chart, and check their

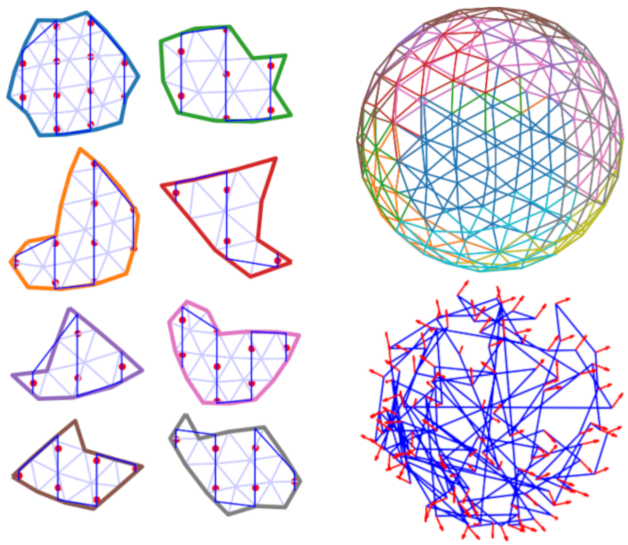

Fig. 1. (Left) A subset of charts with survey paths shown in blue and survey points as red dots. (Right) The final survey path, shown with and without the spherical geometry to survey. The sphere geometry is colored based off of the assigned chart and correspond with the chart outlines on the left.

distortion. Distortion is measured as the fractional change in facet area when projected, itself related to the angle between the chart and facet normals. If the distortion is below some threshold, we add the facet to the chart and recurse until all neighboring facets to the chart would be too distorted. Fig. 1 shows an example chart decomposition for the interior of a sphere.

B. Boustrophedon Planning

Planning a path on the chart is reduced to a 2D coverage problem. We therefore perform a standard Boustrophedon decomposition to generate a path in chart-space. We pick the spacing based on the desired image resolution and overlap percentage. Once this path has been generated, we discretize it to a series of waypoints using the same spacing distance. However, these waypoints still have an intrinsic ordering based on the underlying path.

C. Backprojection and Refinement

We can invert the chart mapping—it is one-to-one by construction—to find the 3D points on the mesh corresponding to our survey points. We can additionally attach a survey direction to them based off of the chart normal. However, these points are on the mesh itself, and we actually want the viewing point for the survey point. Inspired by [3], we use the normal direction to stand-off the viewing point from the mesh surface. There is no guarantee, however, that these viewing poses will be safe (far enough from other obstacles). We therefore perform a final refinement step by computing the nearest mesh surface to the viewing pose. If this distance is less than a given threshold, we push the point away from that surface and repeat until all viewing points are sufficiently far away from the wall. If a point is still not safe after a given number of iterations, it is deleted.

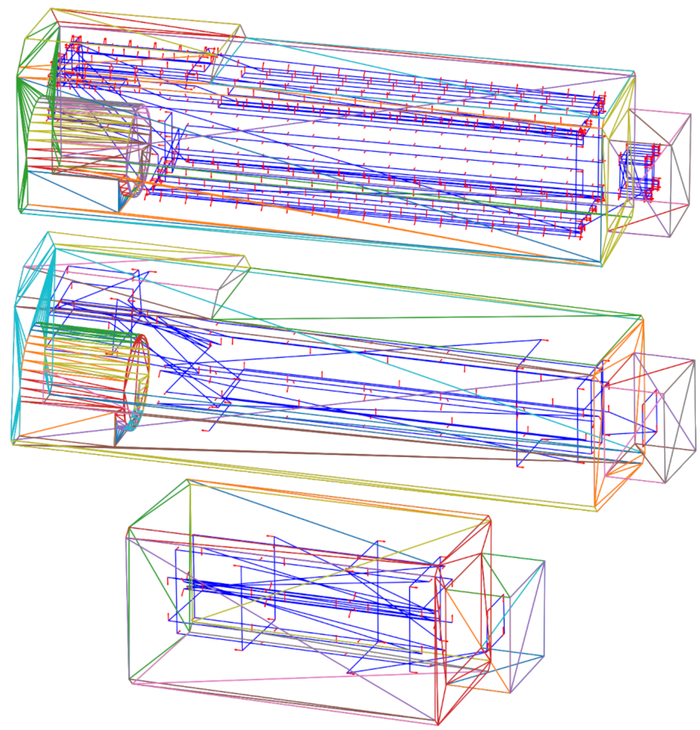

Fig. 2. (Top) JPM high resolution survey, (Middle) JPM low resolution trajectory, (Bottom) Columbus low resolution trajectory.

III. Results

We show survey results for the Japanese Pressurized Module (JPM) and European Columbus Lab in Fig. 2. Note that the survey trajectory naturally follows the main axis of the modules. The effect of varying desired resolution can also be seen in the robot moving further from the wall with greater space between passes when high resolution is not needed. We have also successfully executed these trajectories in the Astrobee simulator, verifying that they are collision-free and executable.

References

[1] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258 – 1276, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092188901300167X

[2] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and Service Robotics*, A. Zelinsky, Ed. London: Springer London, 1998, pp. 203–209.

[3] P. N. Atkar, H. Choset, A. A. Rizzi, and E. U. Acar, "Exact cellular decomposition of closed orientable surfaces embedded in r^3," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 1, 2001, pp. 699–704 vol.1.

[4] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, "Asymptotically optimal inspection planning using systems with differential constraints," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4126–4133.

[5] P. Kafka, J. Faigl, and P. Váňa, "Random inspection tree algorithm in visual inspection with a realistic sensing model and differential constraints," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2782–2787.

[6] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least Squares Conformal Maps for Automatic Texture Atlas Generation," *ACM Transactions on Graphics*, vol. 21, no. 3, p. 10 p, 2002, special issue : Proceedings of ACM SIGGRAPH 2002. Article dans revue scientifique avec comité de lecture. internationale. [Online]. Available: https://hal.inria.fr/inria-00100754