Area Coverage Planning for Sub-dividable Framing Instruments

Russell Knight

Jet Propulsion Laboratory, California Institute of Technology
e-mail: knight@jpl.nasas.gov

Abstract

Scheduling area coverage operations for orbiting spacecraft can be tedious and developing efficient schedules can take time. Many spacecraft have the ability to image sub-sections of areas into tiles that can subsequently be combined. We show that if we leverage this capability, then a fairly straightforward technique can schedule coverage operations optimally and tractably. We apply our approach to imaging glacial areas as part of the Mission to Understand Ice Retreat mission concept analysis.

1 Introduction

Previously we have reported on the Eagle-Eye architecture for Earth-orbiting observation [5]. In this architecture we have a framing instrument that is affixed to a bus that can be moved (either by gimbal or by turning the spacecraft, depending on the specific flight architecture selected) as well as a mirror or other device that allows for small but fast observation of adjacent areas along the boresight of the telescope. Here we focus on planning *area coverage* observations for such a system. Note that this technique is valid for any instrument as long as the focal plane field of view is dividable into halves along each dimension.

Our primary observation is that the telescope boresight can slew slowly while the mirror system maintains pointing on a specific field of view. This allows us to continually gather data (much like push-broom instruments) without needing to slew or settle, increasing the overall efficiency of the data collection. And since the telescope field of view is much larger than the imager field of view, we can divide the areas to be imaged (to a finite extent.) The approach we describe here requires us to divide the telescope fields of view into quarters, half along each dimension, and then produce a telescope imaging path consisting of an ordering of these quarters. This is equivalent to finding a Hamiltonian path on a grid-graph.

2 Approach

The technique is straightforward. After selecting a set of tiles that represent a desired coverage, divide the tiles and link each in concentric paths. Then stitch the paths together at the first availability. We prove that there always is a concentric path and that there always is a legal "stitching point" between each adjacent concentric path.

2.1 Algorithm

First, we assume that the area to cover is decomposable into a set of tiles that result in complete or adequate coverage, thus the input to the system is a set of tiles. Figure 1 shows an example set of tiles to be collected. The goal of finding a coverage path for the tiles is equivalent to finding a Hamiltonian path on a grid-graph.

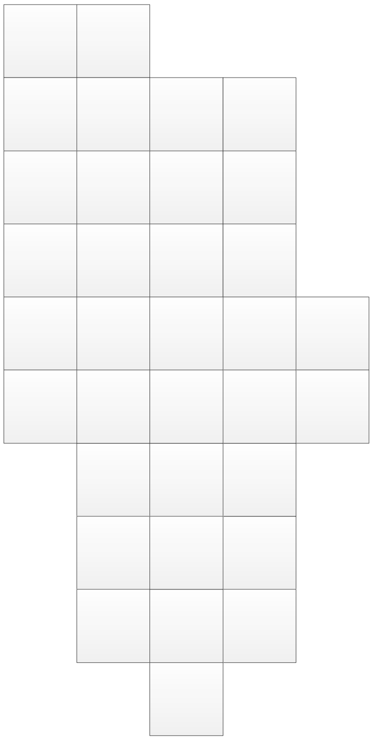

Figure 1. Tiles to be collected

Second, we assume that these tiles can be subdivided both horizontally and vertically into subtiles. Figure 2 shows our example collection of sub-divided tiles. We call the induced grid-graph an ***even grid-graph*** in that all rows and columns have even numbers of elements and all contiguously vertical or horizontal adjacencies contain an even number of vertices.

Given this input, our goal is to select a path through the subtiles that visits each tile once and transitions from one tile to another either vertically or horizontally. In practice, other transitions are allowed, but considered suboptimal.

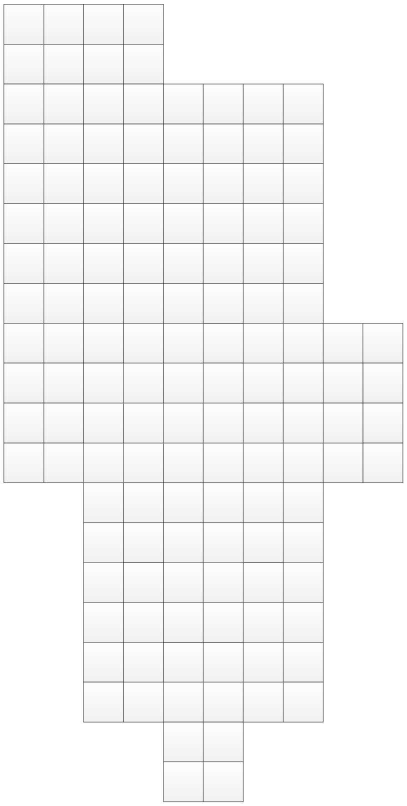

Figure 2. Sub-tiles to be collected

Let’s call this resultantOur approach provides paths that require no other types of transitions. We note that this is equivalent to finding a Hamiltonian path in an even grid-graph.

Formally, given a grid-graph $G = \{V, E\}$, select a path from E that visits each $v \in V$ once. The size of the path is $|V|$ - 1. For notational simplicity, let’s consider each $v \in V$ to be a cell that can be indexed, e.g., the existence of $v_{i,j}$ and $v_{i+1,j}$ would imply an edge in E between $v_{i,j}$ and $v_{i+1,j}$, thus we need not consider the set of edges E and refer to adjacencies using index notation.

Since we are using a grid graph that is the result of subdividing tiles, each $v_{i,j}$ where i is even (i **mod** 2 = 0) and j is even implies the existence of vertices $v_{i+1,j}$, $v_{i,j+1}$, and $v_{i+1,j+1}$.

We start by first decomposing the graph into a collection of cycles by iteratively constructing cycles from the outside-in. Each cycle is very simple to construct: choose a new label, find any un-labeled vertex that has at least one free adjacency (on the perimeter), and follow the cycle around the perimeter until it closes. Remove the cycle once it is completed, and cycle through the rest of the vertices. Figure 3 shows how our collection of sub-tiles can be decomposed into concentric cycles, and Figure 4 shows the final, labeled cycles after decomposition. In section 2.2, we prove that a cycle always exists.

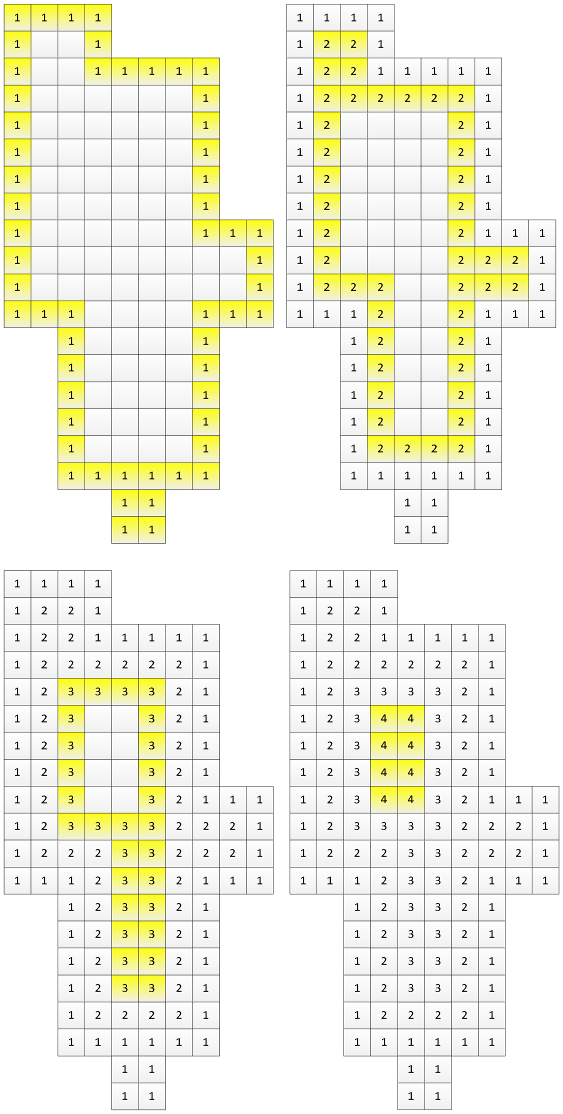

Figure 3. Decomposition into cycles

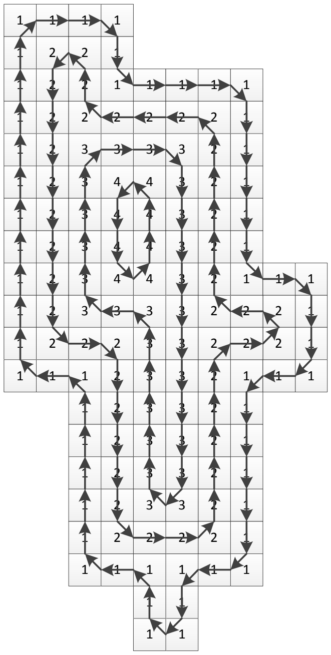

Figure 4. Final labeled cycles

Once we have a collection of cycles, we need to stitch these together to form a path. Again, the procedure is straightforward: select pair of adjacent vertices on the outermost path. This is the entry point. One of the vertices is the first vertex in the path and the other is the last vertex. Then, choose any vertex in any cycle that has not yet been included in the solution path that is adjacent to a vertex that is in the solution. Follow the new cycle until two cycle-adjacent vertices v_1 and v_2 are found that are also grid-graph-adjacent with path-adjacent vertices w_1 and w_2. This is the stitching point. (In section 2.2 we show that there is always a stitching point.)

Figure 5 shows how we iteratively (from the outside to the inside) stitch each cycle to the adjacent outer cycle. The bottom left path is the solution path.

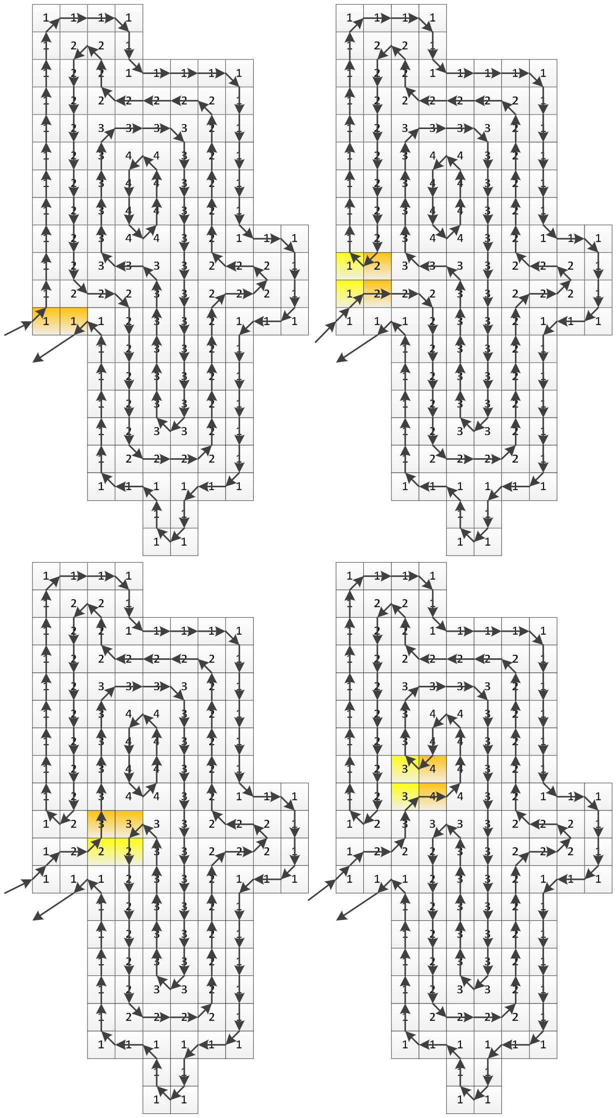

Figure 5. Stitching points after stitching. Orange indicates inner cycle adjacent vertices (before stitching) and yellow indicates outer cycle vertices (before stitching)

2.2 Proof outlines

Here we provide outlines of proofs meant to convince the reader of the completeness, correctness, and optimality of our technique.

First, the problem at hand is the composition of a Hamiltonian cycle for an even grid-graph.

Completeness (that all adjacent vertices are included in any solution) is addressed in that 1) no empty tile goes unlabeled, 2) all labeled tiles are part of some cycle that shares the same label, and 3) all cycles are stitched to containing cycles. If correctness holds, then completeness also holds.

Correctness requires 1) that regardless of how many cyclic decompositions occur, the existence of an unlabeled vertex implies the existence of an as yet unlabeled cycle, and 2) that any two adjacent cycles have stitching points between them.

Lemma 1: *removing the outermost cycle from an even grid-graph results in an even grid-graph.*

Regardless of topology, all columns and rows contain even numbers of vertices. A cycle always removes symmetric vertices (top and bottom, right and left) of any contiguous series of vertices, thus the number of vertices in any column or row remain even. Since the defining characteristic of even grid-graphs is that all columns and rows have even numbers of vertices and all contiguous adjacencies aligned horizontally or vertically have even numbers of vertices, then the resultant grid-graph is even.

Lemma 2: *each cycle has at least 4 stitching points to a containing cycle.*

Because each cycle is constructed from an even grid-graph, the smallest would be a 4x4 graph. In this case, there exist 4 possible stitching points. If we add more adjacent cells, there will always be at least one top, left, right, and bottom cell. Because the cycles offset each other by 1 vertex both vertically and horizontally, it isn't possible to align a "closed center" (an area where there is no gap but is covered by vertices of the same cycle on both sides) from one cycle with an adjacent cycle, thus there will always exist at least one available stitching point along the top, bottom, left, and right of the cycle.

Correctness follows from Lemmas 1 and 2.

The asymptotic size of the problem is the number of vertices in the problem (the number of edges is a constant factor of the number of vertices). Gathering cycles is linear in the number of vertices, as is finding and applying stitching points, thus the overall asymptotic complexity is linear in the problem size. Since we need to evaluate all vertices to produce a solution, the fastest any algorithm could perform is linearly, thus our solution is asymptotically optimal.

3 Application

We have implemented our approach as part of the Eagle Eye architecture. The Eagle Eye architecture is a combination of two automated planner schedulers for the specific purpose of performing observation planning operations for airborne and spacebourne collection systems. The two planner/schedulers are the ASPEN (Activity Scheduling and Planning Environment) [3] and CLASP (Compressed Large-scale Activity Scheduling and Planning system) [6]. ASPEN models provide the interface for adapters. CLASP is leveraged for its ability to calculate coverage statistics on orbital bodies and for its ready interface to the SPICE library [1], which provides spatial and orbital reasoning services.

We have adapted the Eagle Eye architecture to the MUIR (proposed Mission to Understand Ice Retreat) domain [5]. MUIR's mission concept is to mount a telescope on the ISS (International Space Station) and image various ice phenomena (glaciers, ice-sheets, etc.) to determine the change in ice volume over time as well as measure various factors that might contribute to this change. One scenario for MUIR is the imaging of a glacier during a pass, and one concept for MUIR included a mirror that could make small adjustments to the pointing of the telescope, allowing us to choose closely adjacent areas to image (thus providing us with the ability to subdivide our tiles.)

The efficiency of the approach allows us to scan a 3 kilometer by 15 kilometer irregular region during a single pass, as opposed to approaches where we slew and collect, which only allow for a 3 kilometer by 4 kilometer region.

4 Conclusions

We have described a technique that leverages the flexibility of the instruments modeled by the Eagle-Eye architecture to accomplish tiling campaigns to collect data covering contiguous areas of interest. The technique is tractable (linear in the number of induced tiles) and optimal.

Acknowledgement

(c) 2013 All rights reserved. Portions of this work were performed by the Jet Propulsion Laboratory, California Institute of Technology under contract with the National Aeronautics and Space Administration. Clearance number 13-5408.

References

[1] C. Acton, "NAIF, Planetary Data System Navigation Node: SPICE." 31 March, 2014. Web 18 April, 2014. <http://naif.jpl.nasa.gov/naif/>

[2] E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell. "Approximation algorithms for lawn mowing and milling." Technical report, Mathematisches Institut, Universität zu Köln, 1997.

[3] G. Rabideau, R. Knight, S. Chien, A. Fukunaga, A. Govindjjee, "Iterative Repair Planning for

Spacecraft Operations in the ASPEN System." Int. Symp. On Artificial Intelligence, Robotics and Automation in Space (iSAIRAS), Noordwijk, The Netherlands, June 1999.

[4] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter. "Hamilton paths in grid graphs." SIAM J. Comput., 11:676–686, 1982.

[5] R. Knight, A. Donnellan, and J. Green. "Mission Design Evaluation Using Automated Planning for High Resolution Imaging of Dynamic Surface Processes from the ISS." International Workshop on Planning and Scheduling for Space 2013 (IWPSS 2013), Moffett Field, CA, March 2013.

[6] R. Knight and S. Hu. "Compressed Large-scale Activity Scheduling and Planning (CLASP) Applied to DESDynI." Proc. Of the Third IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT 2009), Pasadena, CA, July, 2009.

[7] W. Lenhart and C. Umans. "Hamiltonian cycles in solid grid graphs." Proc. 38th FOCS (1997) 496-507.