Modifying UAV Flight Paths to Build Time-Variant Maps (Extended Abstract)

L. Pinero-Perez[1]
Bell Flight, Arlington, TX, 76019, United States

A. Pai,[2] and M. Schultz[3]
University of Texas at Arlington, Arlington, TX, 76019, United States

The software developed adjusts the existing flight routes of a potentially deployed Unmanned Aerial Vehicle fleet (UAV fleet) to collect useful data in the context of building time-variant map visualizations. With this software, UAV fleet operators can also arrange their flights which maximize the data collection of a region over time. This is a useful by-product of frequent, flexible flights over a populated area. Detailed cyclic time-histories of traffic, populations, or air pollution can be generated by implementing this flight path modification.

I. Nomenclature

δ	=	Related to Deviation from Original Mission Route
N	=	Number of measured units in survey
P	=	Perimeter of a given region on surface
T	=	Total Time of Mission
t	=	Time
v	=	local velocity
x	=	Geography coordinate axis (East and West)
y	=	Geography coordinate axis (North and South)

II. Methodology

A. Initialization

The problem is first addressed by building two time-variant maps: the measured, known map, and the true, unknown map. These are constructed as three-dimensional matrices, composed of orthogonal extrusions into the time-domain of the geographical area evaluated, as shown in Figure 1 on the following page. In application, the length of this time-domain extrusion should be a value related to cyclical human activity, such as a day or a week, so that the data gathered can represent information useful for future predictions. By doing this, consistent departure and arrival cycles are established within the matrices.

[1] Engineer I, Control Laws, AIAA Member.
[2] Student, Department of Computer Science and Engineering.
[3] Student, Department of Computer Science and Engineering.

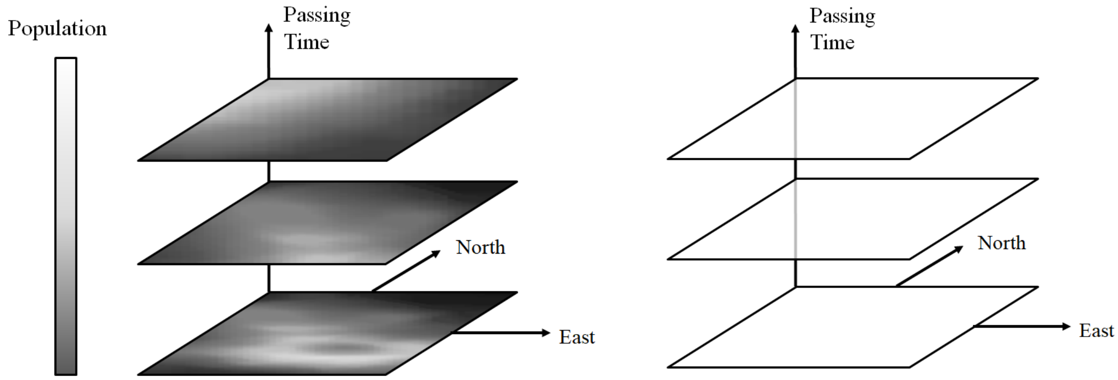

Figure 1. Reality to be Surveyed (Left) and Initialized Map just before Data Collection (Right)

The departure and arrival locations are dispersed as a normal distribution centered on the geographical center (emulating a city center) and regularly dispersed in the time domain (scheduled flights). Each of these locations has a chance to spawn a UAV to connect with another location at a future point in time (i.e. a coordinate further along the time axis). Figure 2 demonstrates how an individual UAV's day might look like when travelling along this matrix.

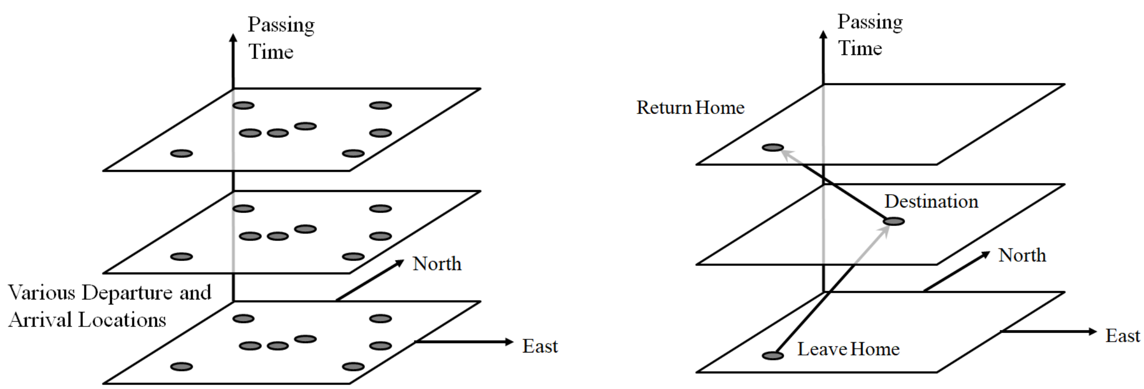

Figure 2. Possible Arrival/Departure Points (Left) and a Possible Route (Right)

B. Unit Flux at Geographical Boundaries

If the observed geographical region is somewhat interconnected with the outside world, there will be a slight unit flux at the geographical boundaries of a measured unit. Where in this paper, the two-dimensional flux is defined as:

$$Flux = \frac{\Delta N}{\Delta t\ (P)}$$

More concretely, this can be a change in the total traffic, population, or air pollution in the region over time (given that these regions are not completely isolated). In reality, this flux should be fairly small, provided that the region's contents are kept relatively whole. "Keeping the region whole" would involve encapsulating a metropolitan area, with the only flux being due to inter-city travel, assumed to be a small proportion of the total population.

However, upon initial surveying, limited UAV data will find that there are large jumps in the total flux of its own smaller perimeter. As stated, this total is found by summing the total amount of data observed by the UAV at each time frame. Initially, the flux along the boundaries of the initially surveyed UAV area should be much larger than the flux at the total boundary. This is because the flux across the observed boundary (driven by the fast moving

UAV's) is much higher than that of the matrix boundary. This difference between the UAV unit flux and the true geographical region unit flux can then be used as a measure of progress in generating a comprehensive census of the measured unit.

C. Velocity Limitations

The velocity time-history of a given UAV can be found by computing the magnitude of the local partial derivative as a function of time.

$$v = \sqrt{\left(\frac{\partial x}{\partial t}\right)^2 + \left(\frac{\partial y}{\partial t}\right)^2}$$

This variable is given saturation, as there are upper and lower limits to the travel of a given UAV. To allow for logistical flexibility, the derivative is kept slightly below saturation. This setting is what will allow for curved paths along the space vs. time matrix in the next section.

D. Route Modification

The routes are now ready to be modified to maximize the amount of volume coverage. This will involve changing the UAV velocity and/or deviating spatially from its shortest path over time (in the context of its original mission). Figure 3 shows how a single drone can increase its coverage over the course of multiple flights.

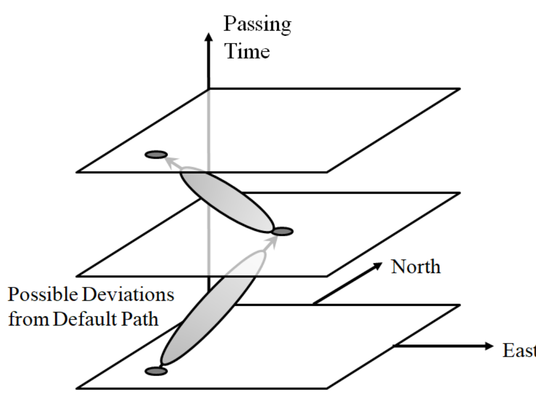

Figure 3. Modifying the Possible Route

The optimal system will be evaluated on its ability to allocate existing routes so that the UAV's do not have to be significantly deviated from their original routes. Such a deviation could be considered a cost to be minimized, where the cost can be determined as:

$$\delta = \int_0^T \left(\sqrt{(v_\delta - v_{mission})^2 + (x_\delta - v_{mission})^2 + (y_\delta - y_{mission})^2}\right) dt$$

The software will maximize the amount of volume covered over time, as more and more flights take place. The efficiency of such a system's coverage must be normalized according to fleet size and operational flight frequency.

E. Evaluation

At the end of simulation, the path optimization system will also be evaluated according to the volume coverage of the total time-map. While data values collected within the covered area do not contribute to the total coverage performance, the flux can be used to give a measure of progress in the context of the goals of a survey. For example, if the total number of sheep were being tallied, it would not matter if there were swaths of time and space which have not been observed, so long as a significant proportion of the population is captured. The flux can be a good measure for meeting survey goals so long as there is not a large group that is consistently avoiding detection in this matrix.

F. Visualization

Time-variant map visualizations created from the data collected by the UAV fleet are accessible to the user through an offline-first, IndexedDB-based visualization dashboard running through a browser window, similar to Jupyter Notebook or Tensorboard and through a Python wrapper for developers. The visualization dashboard works as a web application with a Flask server on the backend along with socket.io for real-time bi-directional synchronized communication between the client-facing part of the dashboard and the server side. The dashboard essentially provides a graphical user interface (GUI) for users to make changes to parameters and iteratively test to optimize flights to maximize data collection.

A few different libraries are being considered for the generation of different visualizations based on capability and user familiarity. Given the Python backend to this visualization tool, Matplotlib is one of the main libraries being considered along with Seaborn for producing more appealing statistical graphs generated alongside the time-variant maps. For the time-variant maps specifically, the Geoplotlib library is being considered for its usefulness when it comes to visualizing geographical data. The three-dimensional version of the time-variant maps is visualized as possibility cones (defined by max-range radius and travel time) and Basemap 3D from the Basemap library from the matplotlib mplot3d kit seem like the right choice for that visualization. Some more web application-friendly visualization libraries are also considered for more seamless updating. Pygal, for its interactive capabilities, and Bokeh's real-time data streaming, and JSON exporting capabilities might be part of the final product.