Learning to Improve Earth Observation Flight Planning

Robert A. Morris[2] and **Nikunj Oza**[2] and **Leslie Keely**[2] and **Elif Kürklü**[1,2]
(1) Perot Systems Government Services
(2) Intelligent Systems Division
NASA Ames Research Center

Abstract

This paper describes a method and system for integrating machine learning with planning and data visualization for the management of mobile sensors for Earth science investigations. Data mining identifies discrepancies between previous observations and predictions made by Earth science models. Locations of these discrepancies become interesting targets for future observations. Such targets become goals used by a flight planner to generate the observation activities. The cycle of observation, data analysis and planning is repeated continuously throughout a multi-week Earth science investigation.

Introduction

Machine learning has been integrated with planning systems in order to automatically extract knowledge from one problem to apply to future problems, thereby improving the performance of the planning system. More specifically, knowledge about the domain, search control strategies, or solution quality, can often be refined or extended from the ”experience” of planning.

The work described in this paper fits into the overall idea of ”learning to improve planning” but with an emphasis that distinguishes it from the goals of other efforts. Specifically, the ”experience” that improves the planning is gained from (data acquired as the result of) the *execution* of previously generated plans, rather than the experience of planning (i.e. the search for a plan) itself. The application here is daily mission flight planning from the identification of useful or interesting observation targets, for an in-situ sensor mounted on an airborne observatory (a modified DC-8). Typically, targets are interesting if they reveal discrepancies or anomalies in predictive models used by Earth scientists to study processes related to things like climate change or pollution. Further observations of such targets will quantify the errors in the predictive models in order to improve their predictive accuracy or more generally fill gaps in the knowledge about the process of interest. Interesting observation targets become goals used by a flight planner to generate the next day's observation activities. Furthermore, because the process under investigation (e.g. a hurricane or pollution plume) may be changing daily, a *cycle* of planning, observation, analysis of data acquired from observation, and model revision and prediction, occurs continuously throughout the course of an *observation campaign* (typically lasting weeks). Finally, to be fully effective as a mission operations tool, it is necessary to have an effective way to visualize both the data being analyzed and the plans being generated.

This paper presents an early prototype of a system for integrating planning, learning and visualization capabilities for use by human mission planners to improve the daily operations of what are called *mixed observation* missions, i.e., missions that combine observations from a number of heterogeneous ground, airborne, or space-borne sensors. The next section offers a detailed example of a target mission and an overview of the integrated system. There follow sections on the learning, planning and visualization capabilities required.

Application and System Architecture

NASA and other international space agencies launch and operate Earth observing systems for collecting remote sensing measurements to support scientists in the pursuit of goals related to understanding changes to the Earth's ecosystem. These data are combined with data collected from *in-situ* sensors mounted on airborne or ground platforms. Many scientific goals require the combination of data acquired from different sensors. For example, to expand the temporal and spatial scale of airborne measurements, measurements from remote sensors are obtained simultaneously with in-situ sensors.

An example of such a mixed platform observation mission was INTEX-B (INTEX-B), conducted from March 1 to April 30, 2006. INTEX-B was the second part of a two-phase experiment that aimed to understand the transport and transformation of gases and aerosols on transcontinental and intercontinental scales and assess their impact on air quality and climate. INTEX-B science goals include quantification of the outflow of pollutants and aerosols from North America over the Atlantic to Europe; an improvement of our understanding of the chemical and physical evolution of these constituents; and an assessment of the impact of pollution transported from mega-cities such as Mexico City. High-priority measurements included long-lived greenhouse gases, ozone and its precursors, and aerosols and their precursors.

Day-to-day DC-8 flight planning on INTEX-B was a process of generating a set of flight legs between waypoints (the flight plan). Plan generation was partly automated using various flight planning tools that assist the human user by accounting for flight dynamics of the DC-8 and other constraints. Selection of waypoints was guided by forecasts generated from a set of models of different spatial scales. Near-real-time observations from a number of satellite instruments were used to guide the selection of and to identify specific regions of interest for in-situ sampling. Integration of aircraft and satellite measurements to address the mission objectives requires validation flights directed at establishing the consistency between the two data sets.

The focus of automation in the context of mixed observation missions such as INTEX is to improve the ability to *assimilate* the different data sources in order to generate waypoints; and secondly, to automate the *search* for high quality plans, where quality is related to the ability to resolve model anomalies or accomplish other mission goals. The idea is that improved data assimilation will result in a better understanding of the space of high quality flight plans, a space which can be then explored more effectively by automated planning methods.

The semi-closed system integrating machine learning with flight planning into a daily mission planning system is visualized in Figure 1. The closed loop system involves four components:

1. Data archives, process models, and instruments for producing data or forecast products (commonly referred to as a ”sensor web”).
2. A *data assimilation assistant* for generating planning goals (waypoints) given data;
3. A *flight planning assistant* for generating flight plans given goals; and
4. A *visualization tool* for displaying the data and plans to the human flight planner.

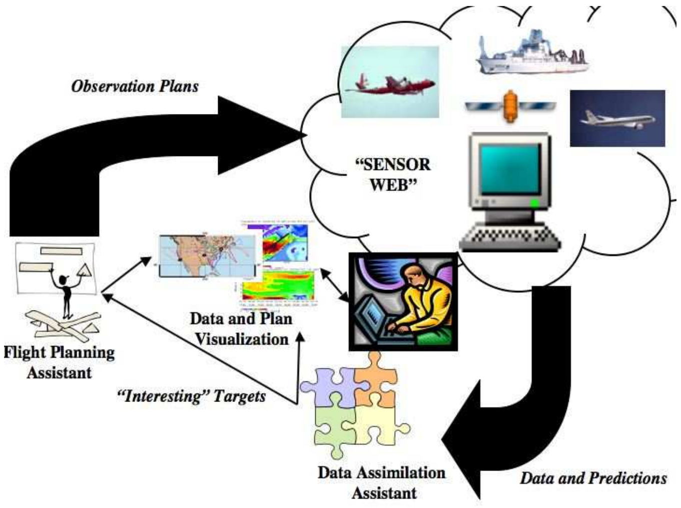

Observation Plans
"SENSOR WEB"
Flight Planning Assistant
Data and Plan Visualization
"Interesting" Targets
Data Assimilation Assistant
Data and Predictions

Figure 1: The daily mission flight planning cycle, integrating machine learning, visualization and flight planning.

For example, a process model produces Carbon Monoxide (CO) distribution predictions for a specified region, which are assimilated with satellite observations of CO from remote sensors over the same region, such as Aurora and Ionospheric Remote Sensing (AIRS) and Measurements of Pollution in the Troposphere (MOPITT) sensors, as well as observations of dust, aerosols and clouds from the Moderate Resolution Imaging Spectroradiometer (MODIS). These data and models are organized and archived. As a result of data mining, waypoints, expressed in a 3D location vector, $\langle latitude, longitude, altitude \rangle$ and a value indicating priority, are identified as interesting locations for conducting future CO measurements, either for finding and characterizing differences between models and observations, or for filling in gaps in data. The waypoints are fed into the planner as goals. Flight plans are generated and displayed to the human planner using the visualization tool. Measurements are taken, and the cycle is repeated, usually on a daily basis.

The remaining sections focus on the learning, planning, and visualization challenges that arise in this application.

Learning Problem

As indicated above, the data assimilation assistant generates waypoints for planning. The waypoints are intended to represent locations where the data is most ”interesting.” For example, ”interesting” could mean that instruments intended to measure the same quantity yield significantly different measurements, or that process models yield predictions significantly different from supposedly corresponding measurements, or mathematical relationships expected to hold between various models and measurements do not hold. So far, we have mainly experimented with the second metric—differences between model predictions and measurements—and will report on these in the limited space that we have. We have done some preliminary investigations into the third metric—mathematical relationships between models and various measurements—however, we need to experiment further with selecting the right subsets of data over the right time periods and adjusting appropriate tuning parameters to obtain a reasonable set of waypoints.

We compared a model prediction of carbon monoxide (CO) with two satellite measurements of CO. The process model that we used was the Model of OZone And Related chemical Tracers (MOZART) (Brasseur *et al.* 1998). MOZART predictions were compared with data acquired by the AIRS and MOPITT remote sensors. The MOZART predictions were placed in a regular coarse grid (with each grid cell covering 0.7059 degrees latitude by 0.7031 degrees longitude). The AIRS and MOPITT instruments take their measurements in swaths because of the satellites' orbits around the Earth (the swaths can be compared to a ribbon being wrapped around a sphere). Therefore, the measurements end up in data products as many measurements at various points within a swath. We then selected the points in all three datasets that fell within the region of interest—between 12 and 32 degrees north latitude and between 80 and 105 degrees west longitude—which is Mexico and the surrounding area. Altitude-wise, we selected the portions of AIRS and MOPITT that fell within the range present in MOZART—

from the Earth surface to approximately 43 kilometers. We ended up retaining 131474 AIRS measurements and 14250 MOPITT measurements.

For each of these AIRS and MOPITT measurements, we then found the closest MOZART point and calculated the difference between the CO measurement and the MOZART CO prediction in units of parts per billion. We then converted the resulting 131474 AIRS-MOZART differences into priorities on a scale of 1 to 10 for input into the planner by simply linearly scaling the absolute differences into this range. We retained the 50 points with highest priorities such that the points were a minimum of 58 kilometers (approximately 10 minutes of DC-8 flight time) apart. We did this because the instrument had the constraint that it needed at least 10 minutes of measurement time around a waypoint in order to obtain a clear measurement. Planning for multiple waypoints that are close to one another while still satisfying the time constraint would have required a flight plan with excessive turns. We selected the 50 points simply by first selecting the highest priority waypoint, removing all the waypoints less than 58 kilometers away from this highest-priority point, selecting the remaining waypoint of highest priority, removing all the waypoints less than 58 kilometers away from this waypoint, etc., until we obtained 50 points. We separately repeated this process of selecting 50 waypoints from the 14250 MOPITT-MOZART differences. We restricted ourselves to 50 waypoints to limit the planner's execution time. If we had only retained the constraint on minimum distance between waypoints, we would have had 302 AIRS-MOZART waypoints and 91 MOPITT-MOZART waypoints.

One could argue that the Flight and Activity Planning Assistant (FAPA) should perform the selection of waypoints based on the minimum distance between waypoints, since this is an instrument-related constraint. This way, there would be a clear division of responsibility with the Data Assimilation Assistant (DAA) processing data, and the FAPA dealing with flight and instrument-related constraints. Our decision here was based purely on efficiency—because the waypoint selection function is distinct from the planning algorithm, and the DAA produces the waypoints and has them in memory, selecting the appropriate waypoints here is faster.

We also experimented with standard unsupervised learning methods for clustering (K-means (MacQueen 1967) and kernel clustering (Girolami 2001)) and anomaly detection (one-class Support Vector Machines (Tax, Ypma, & Duin 1999)). Clustering methods organize the data into clusters that represent data with similar properties (e.g., particular ranges in which different measurements fall or particular mathematical relationships between measurements). Data that are part of very small clusters or are far from all the clusters represent possible anomalies. Anomaly detection methods construct a model of the training data, which is typically assumed to be normal. These methods are then executed on new data, and data that do not fit the model's definition of normality are flagged as anomalous. For example, one-class Support Vector Machines (SVMs) are given training inputs $z_1, z_2, \ldots, z_n$, as well as an expected fraction of training points that are anomalous. One-class SVMs find a nonlinear model that identifies up to that fraction of training points as anomalous and separates them from the normal points. When a one-class SVM is given a new point z, it returns a value y that is positive if z is part of the normal regime identified in the training data and negative if z is thought to be abnormal. The farther away y is from zero, the "more normal/abnormal" z is.

However, assessing the points returned by these methods to determine if they represent true anomalies is more difficult than assessing the points with high differences as we have mainly done so far. The differences are measured in units familiar to domain experts (parts per billion), whereas distances from cluster centroids (for clustering methods) and distances from hyperplanes (for one-class SVMs) do not have clear interpretable semantics. The domain expert may have to directly examine the points flagged by these more sophisticated methods in the hope of being able to figure out the nature of the anomalies that the methods have flagged. This has limited our experimentation with clustering and anomaly detection methods so far. Additionally, we need to experiment with different subsets of data products covering different periods of time and with tuning parameter adjustment in order to yield a reasonably large set of waypoints. In spite of these difficulties, this work is ongoing because of these methods' ability to characterize anomalous relationships between variables even though the variables themselves may be in the correct range. We expect that these methods will find waypoints not found using methods typically used by domain scientists.

Planning Problem

There are two kinds of inputs to the flight planner: mission goals and waypoints. Mission goals arise from models, from a phenomenon or event of interest, or from other sensing resources. The goal might be to validate predictions made by a model, to characterize or classify the composition of dynamic process like a pollution plume, or validate observations made by a remote sensing instrument. The waypoints are 3D specifications of locations that support one or more of the goals, generated from the data assimilation assistant just described. Goals or waypoints are assigned priorities indicating importance.

The output of the flight planner is a flight plan, a closed path comprised of a set of segments, or flight legs, connecting waypoints. A path is also associated with temporal information (start time of each leg, and duration). Each leg follows a pattern: direct ascent, direct descent, level, or spiral (ascent or descent). The best path is the one the planner has determined most likely to satisfy the most goals, based on the specified priorities.

There are four kinds of constraints on solutions generated by the planner.

1. *Instrument Constraints.* Instruments must be set up before they can take measurements. This means two things: for the aircraft to assume a certain flight pattern, and to remain in this pattern for a specified duration of time.

2. *Aircraft Constraints.* The DC-8 has flight requirements related to its navigational capabilities, or for safe flying.

3. *Path Constraints.* Flight paths must satisfy certain logistical constraints, such as starting and end at the same location. Another important constraint of this type is the avoidance of Special Use Airspace (SUA).

4. *Mission Goal Constraints.* Certain flight segments are required by the goals of the missions themselves. Typically these goals involve inter-comparison of data acquired by different sensors. For example, as noted, on INTEX, the data acquired by the in-situ sensor on the DC-8 was often compared to remote sensing data acquired for the same region at the same time, so the DC-8 was required to fly along the same track as that flown by the remote sensor.

The core computational problem being solved by flight planning can be viewed as the *orienteering problem* (OP) (Fischetti, Gonzalez, & Toth 1998). Formally, given a set of waypoints, $W = \{x_0, x_1 ... x_n\}$ each with an assigned priority $w_i, 1 \leq i \leq n$, a binary cost function $C : W \times W \rightarrow N$, a designated start point x_0 with $w_0 = 0$, and a temporal bound B, find a sequence (schedule) $s = \langle x_{s_1}, ..., x_{s_k} \rangle, s_j \in \{0, \ldots, n\}, k \leq n+1$, that maximizes $\Sigma_{j=1 \ldots k} w_{s_j}$, subject to the following constraints:

1. $s_1 = s_k = 0$,
2. $i \neq j \rightarrow s_i \neq s_j$, i.e., each $x_j \in W$ occurs at most once in s,
3. $\Sigma_{i=2 \ldots k} C(x_{s_{i-1}}, x_{s_i}) \leq B$.

Notice that in this version there is an assumption of "over-subscription": it is not necessary, and may not be possible, to service all the customers, and indeed a best route might not include all of them.

A constructive approach to flight planning was used, where each decision point involves selection of the next goal to add to a partial schedule. A greedy stochastic approach to extending a partial solution is employed, where each feasible waypoint candidate x_i is heuristically evaluated in terms of the *expected value* v_i of adding a leg terminating at x_i to the partial plan. The heuristic value biases the random selection process in favor of that candidate, similar to the HBSS technique employed in (Bresina 1996). The purpose of injecting non-determinism into the solver is to be able to generate multiple solutions for comparative evaluation by the human planners.

We define $v_i = w_i / [C(x_{s_j}, x_i) + p_{j,i}]$, where x_{s_j} was the last waypoint added to the schedule, and $p_{j,i}$ is a penalty for SUA intrusion. A candidate x_i is feasible for a partial schedule $\langle x_{s_1}, \ldots, x_{s_j} \rangle$ if the sequence $\langle x_{s_1}, \ldots, x_{s_j}, x_i, x_{s_1} \rangle$ satisfies the bound constraint (3), i.e., if the aircraft can return to the airport immediately after flying to the waypoint without violating (3). The algorithm deterministically selects the candidate with the highest expected value to extend a plan, until either the list of available candidates is empty, or none of the remaining candidates is feasible with respect to the current plan.

The flight planning algorithm can be described as follows:

Flight Planner

```
While there are observations left to schedule
    1. Greedily select next observation to schedule:
            For each unscheduled observation
                Evaluate cost of adding the observation to
                the schedule, including the need to avoid SUAs.
    2. Add enabling leg
            Check altitude constraints
            Find path to avoid SUA incursions, if necessary
            Check to ensure resulting schedule does not violate
            flight duration constraints.
    3. Add observation leg (same steps as in 2.)
Add a flight leg to return to origin.
return plan
```

For each waypoint in the input to the problem, the planner computes the expected value v_i, described above, and (non-deterministically) greedily selects the observation with the highest such value. An *enabling leg* is added to the plan, which is the shortest path to get to the observation point. The leg must not violate altitude, duration, or SUA incursion constraints. An *observation leg* is then added to the plan, corresponding to the time of the measurement at the desired location. The same tests for feasibility are made as those made when the enabling leg is added. The planner algorithm is re-run to generate as many plans as desired.

For normal problem sizes of 50 candidate waypoints, a single plan can be generated in roughly three minutes, so it is feasible to generate many plans for comparison. Most of the hard computational effort arises from SUA avoidance. For this, we employed traditional obstacle avoidance techniques based on visibility graphs. SUA avoidance is discussed in detail in (Kürklü, Morris, & Oza 2007).

Visualization

The goal of the visualization component is to provide a three dimensional and interactive spatial context of the mission. This context allows the human flight planner to view and validate the results of automated methods and provides a means to explore and examine flight plans with "what if" scenarios. To meet this goal this component has a number of objectives:

1. represent the relevant data and analysis results within a single geo-registered framework,
2. provide interactive techniques for viewpoint designation, data interrogation, and scene management,
3. provide multi-scale representation with varying levels of detail, and
4. deliver everything to the mission flight planners via an easy-to-learn interface.

Current practice requires comparing many different maps, plots, and other documents at different resolutions and projections. Mission scientists must look at these documents separately and make time-consuming conversions between them. Making all resources available in a single geo-registered frame allows for timely spatial comparison and aggregation of data, as well as validation and exploration of automated learning and planning results.

Resources used for the INTEX mission include a base map of the Gulf of Mexico, coordinates for special use air space regions, satellite ground track coordinates, 42 atmospheric layers of predicted CO concentration, candidate waypoints from the learning process, and the flight plan from the planning process. The visual frame rapidly becomes cluttered when all of the relevant mission resources are included. Some objects obscure others and some objects become lost in the crowded scene. Interactive scene management (hide/show capabilities) and viewpoint designation provides the capability of changing views of the same scene and designating what parts of the scene are to be displayed. These features are critical in viewing and understanding the spatial relationships between resources.

Important objects in the mission context maintain a number of attributes (for example, waypoints have altitude and priority) that affect the outcome of the learning and planning processes and as such are of interest to flight planners. Access to these attributes in an interactive and spatial manner is very useful for understanding the results of these processes.

As is common with mixed observation missions, interesting features are found at varying scales. Overall flight plans and model predictions are viewed in relation to a small scale map and flight legs are viewed at a larger scale. Zooming and multiple levels of detail are useful for viewing a multi-scale context.

Two visualization tools were used for this work. Mercator, a geo-visualization software application under development at Ames Research Center, was used to create the visualizations of the data and learning and planning results. Google Earth was used to deliver these visualizations for examination. Mercator imported the data of various formats, created visualization objects, and exported them to KML files which Google Earth could then display. Google Earth is a well-known interactive software application for viewing maps of the Earth that is free and easily available from the Google website. It provides the scene management, multi-scale display, and multiple viewpoint designation needed for viewing the spatial context created by Mercator.

A visualization of the Gulf of Mexico was created using a digital elevation model obtained from the Jet Propulsion Lab's OnEarth web map server. This visualization was used in Mercator to provide a reference frame for the other visualizations. Google Earth also provided a base map.

The visualization of the special use air spaces consisted of a set of extruded polygons with a ceiling and floor. Some of these polygons were concave and required tessellation. These objects were made transparent for visibility of the surrounding terrain and output as model files.

The predicted CO of the MOZART model was visualized as a set of surfaces at varying altitudes. CO concentration was indicated by transparency. Relatively opaque areas have higher concentrations.

The satellite ground track was represented by a polygon draped along the width of the track over the base map. This polygon represents the actual width of the satellite track.

Waypoints were indicated by an icon placed at the waypoint location and the priority attribute was indicated by icon size. Waypoints can be interrogated for attributes.

The flight plan was depicted with a set of icons representing the waypoints and lines connecting the waypoints. The lines were tessellated to conform to the contour of the earth.

The visualizations were exported to KML formatted files for input into Google Earth. Each file contained a data set or result. As can be seen in figure 2, users can view a combination of candidate waypoints, SUAs, and the flight plan geo-registered with the base map. Additionally, shown in figure 3, users can drill down and view a flight leg at a greater level of detail. Figures 4 and 5 show the predicted CO visualization in Mercator and Google Earth respectively. Users can select one of the 42 levels with a slider. A plume of higher concentration over Mexico City is visible as a bright spot.

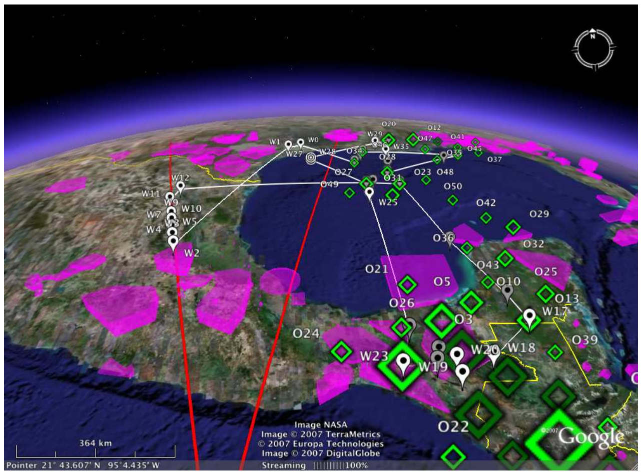
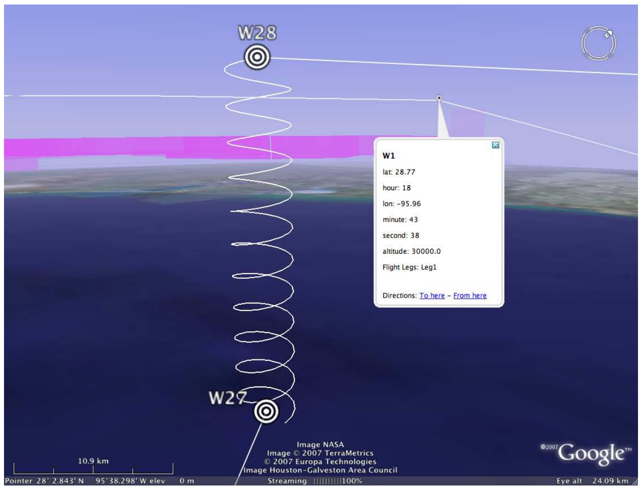


Figure 2: Google Earth with SUAs (magenta), satellite ground track (red), MOPITT/MOZART difference waypoint candidates (green diamonds), and flight plan (white).


W1
lat: 28.77
hour: 18
lon: -95.96
minute: 43
second: 38
altitude: 30000.0
Flight Legs: Leg1

Directions: To here - From here

W28
W27
10.9 km
Image NASA
Image © 2007 TerraMetrics
© 2007 Europa Technologies
Image Houston-Galveston Area Council
Pointer 28° 2.843' N 95° 38.298' W elev 0 m
Streaming 100%
Eye alt 24.09 km
Google

Figure 3: Detail of flight plan showing a spiral leg and attributes for a waypoint.

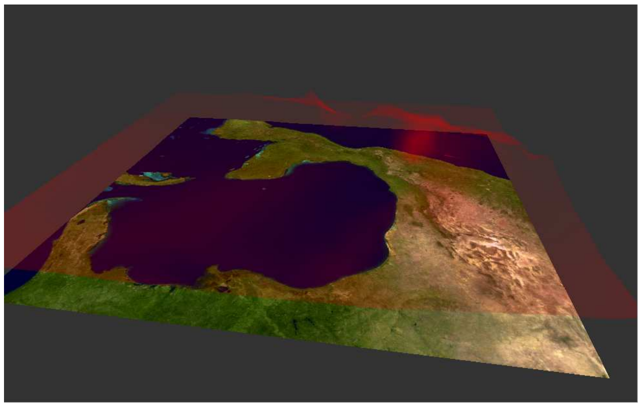


Figure 4: View of MOZART CO prediction at level 27 with 100X vertical exaggeration in Mercator.

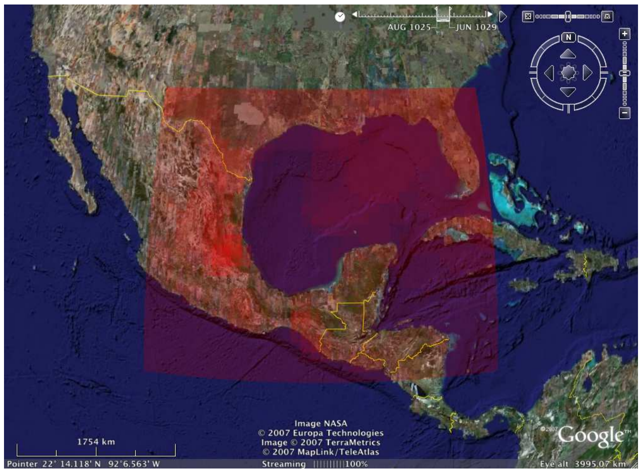


Figure 5: View of MOZART CO prediction in Google Earth.

Current Status and Conclusion

This paper has described a work in progress.The bulk of the effort to date has been on the testing and validation of the component learning, planning, and visualization tools. Integrated testing and validation of plans generated from the assimilation of real INTEX data has only just commenced. Consequently, it is difficult at this point to rigorously assess the utility of this approach. Preliminary feedback from scientists has indicated that

- The visualization capability is of immediate benefit because it enables scientists to compare many different maps, plots, and other documents at different resolutions and projections.
- The data assimilation component is also of immediate benefit because it, combined with the visualization tool, enables quicker identification of discrepancies between model predictions and observations. It needs to be extended to account for more measurement targets, process models, and satellite data.
- The flight planning component is of potential benefit, not so much as a fully automated process, but in a mixed initiative setting to assist human planners in creating optimal plans, as well as to compare alternative plan scenarios. The greedy nature of the planner leads it to sometimes bypass lower priority waypoints that it should cover because they are "along the way," but it nevertheless achieves the goal of covering high-priority waypoints while satisfying flight time and other constraints.

This paper has described an infusion of AI technology into decision support tools for remote sensing missions for Earth science. The innovative use of machine learning and data assimilation to improve mission observation planning will increase the amount of useful data products for improving the predictive capabilities of Earth science models, thus improving human understanding of Earth processes. From a technology standpoint, the use of observation and forecast data in the formulation of planning goals represents a more robust representation of the world in which the plans will be executed, which will improve the ability of planning systems to converge on plans with high scientific value. Finally, the integration of data from models and measurements with planning for atmospheric studies such as INTEX requires a robust three dimensional and interactive spatial context of the mission.

References

Brasseur, G.; Hauglustaine, D.; Walters, S.; Rasch, P.; Muller, J.; Granier, C.; and Tie, X. X. 1998. Mozart: a global chemical transport model for ozone and related chemical tracers, part 1. model description. *Journal of Geophysical Research* 103:28,265–28,289.

Bresina, J. L. 1996. Heuristic-biased stochastic sampling. In *AAAI/IAAI, Vol. 1*, 271–278.

Fischetti, M.; Gonzalez, J. J. S.; and Toth, P. 1998. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing* 10:133–148.

Girolami, M. 2001. Mercer kernel based clustering in feature space. *IEEE Transactions on Neural Networks* 13:780784.

INTEX-B. http://www.espo.nasa.gov/intex-b/.

Kürklü, E.; Morris, R. A.; and Oza, N. 2007. Learning points of interest for observation flight planning optimization: A preliminary report. In *Proceedings of the Workshop on AI Planning and Learning*. International Conference on Automated Planning and Scheduling.

MacQueen, J. B. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1. Berkeley: University of California Press. 281–297.

Tax, D. M. J.; Ypma, A.; and Duin, R. P. W. 1999. Pump failure detection using support vector data descriptions. In *Lecture Notes in Computer Science*, volume 1642. 415+.