CS486C – Senior Capstone Design in Computer Science
Project Description

Project Title:	Optimizing Visualization of Remote Sensing Data in Antarctica
Sponsor Information:	**Dr. Mark Salvatore**, Assistant Professor of Planetary Science Department of Astronomy and Planetary Science Northern Arizona University mark.salvatore@nau.edu **Dr. Heather Lynch**, Professor Department of Ecology and Evolution Stony Brook University Heather.Lynch@stonybrook.edu

Project Overview:

The Antarctic continent is the most isolated continent on Earth and, as a result, hosts some of the world's most pristine environments. Scientists studying a range of natural phenomena, from penguin rookeries to volcanic processes, turn to the Antarctic and its unmodified landscape. However, traveling to and studying in Antarctica is difficult and dangerous, and increased human presence risks altering the pristine nature of this landscape. As a result, scientists have increasingly turned to high-resolution satellite-based investigations as a means of remotely studying this landscape.

The size of Antarctica and the number of satellite images necessary to investigate continent-wide phenomena present their own challenges to remote sensing investigations. For example, nearly 53,000 images from the high-resolution WorldView-2 (WV2) satellite are required to cover the entirety of the Antarctic continent. As a result, the National Science Foundation (NSF) funded the Imagery Cyberinfrastructure and Extensible Building-Blocks to Enhance Research in the Geosciences (ICEBERG) Project as part of the EarthCube Integration Program to design and build code to automatically execute image processing and analysis techniques over large spatial areas. For the past three years, the ICEBERG Project has developed workflows for several focused Antarctic projects, including land cover classification, identifying and mapping glacial streams, and the remote identification of seals and penguins along the Antarctic coast.

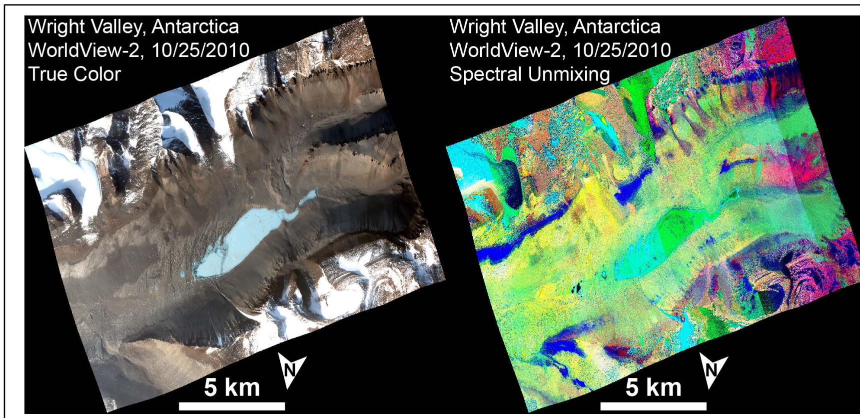

Figure 1. A visible WorldView-2 image of Wright Valley, Antarctica (left), and a derived satellite image product of the same region, with red, green, and blue colors corresponding to sandstones, granites, and dolerites, respectively.

Despite these successes, significant challenges remain. For example, each satellite image contains eight spectral bands that record individual wavelengths of light reflected off of the Earth's surface, which can be used to interrogate surface composition (**Fig. 1**). These bands must be corrected for the effects of the atmosphere, which can distort the image and scatter light being reflected off of the surface. The atmosphere can also frequently change in Antarctica, making it necessary to assess the atmospheric contributions in each individual image being investigated. Once the atmospheric effects are removed, the observed spectral variability (differences in reflected light at each wavelength) is due to variations in surface composition and physical properties. These variations can be interrogated using libraries of field- and lab-derived spectral signatures in order to quantify their individual contributions to the reflected light. Correcting for atmospheric effects, identifying unique surface components, and correlating these components with individual surface materials are tasks that are currently being integrated into the automated ICEBERG pipeline.

Another significant challenge to this work involves the visualization and interrogation of such large volumes of satellite data. While the ICEBERG Project is working on data processing techniques, end-user querying and data visualization are not current areas of emphasis. Therefore, we are looking for a team to develop a user-friendly, interactive, and scalable tool to visualize, navigate, and query raster image products consisting of multiple layers that represent surface compositions, classification confidence, and other information for end-user analyses. The tool would allow for different layers/features to be turned on/off, would have a query tool to investigate the underlying data contained within the images, and would allow for screenshots to be exported and saved locally. Such a product would enable the broader scientific community to use the end-products generated as part of the ICEBERG Project. An example of such a GUI interface might be the Mapping Application for Penguin Populations and Projected Dynamics (MAPPPD) tool that has been developed specifically for analyzing data on penguin populations: http://www.penguinmap.com/mapppd. While the ICEBERG interface would, of course, have completely different focus/features, the MAPPPD tool nicely illustrates some key "look and feel" characteristics: nice graphical UI, efficient and easy to find what you want, and accessible to competent but not "tech savvy" scientist end-users.

Specifically, we envision a secure web application that provides easy access to ICEBERG data and key tools to scientist end-users. Some of the key functions supported by this product should include: :

- Minimum Viable Product: A web-based tool written in Python that can visualize multi-layer float GeoTIFF data, which would include:
  - A GUI interface with easy navigation and zoom functionality;
  - The ability to quickly ingest large (GB-scale) datasets for rapid visualization (i.e., does not require long load times or hinder the ability to navigate between different geographic regions); and
  - A query tool that allows for the easy extraction of data from these raster files.
- Complete Product: A product that would enable all of the above functionality, in addition to:
  - Tools to annotate the visualized data, including scale bars, legends, and north arrows;
  - The ability to toggle between image layers (or separate images) and other data layers (e.g., points;
  - The ability to load multiple *smaller* images (projecting them in proper geographic coordinate space) or individual *larger* image mosaics while maintaining speed and navigability (through tiling or other similar techniques); and
  - The ability to export subsets of the data locally onto the user's personal computer.
- Stretch Goals: A product that would enable all of the above functionality, in addition to:
  - The ability to perform:
    - Mathematical functions on the image data;
    - Data cluster analyses (e.g., the conversion of raster data to grouped vector data); and
    - Statistical analyses on identified image subsets, including mean and standard deviation.
  - Crowdsourcing image interpretations where users can identify clusters of pixels, outline these clusters, and create rough polygons that can be viewed and commented on by other investigators.

This tool would become the standard end-user product attached to the ICEBERG Project, as it would provide a means of quickly interrogating and querying data without the need for high-performance or proprietary software packages. This tool would be applicable to many in the Earth and environmental sciences.

Knowledge, skills, and expertise required for this project:

- An understanding of raster data processing and interpretation;
- Knowledge of modern Web2.0 programming techniques, languages, and frameworks.
- Skills in GUI design and evaluation.

Equipment Requirements:

- A cloud-based server will likely be required as a deployment platform. Development can be done on a free-tier server available from AWS. At product delivery, the client will take over this server, or will provide a final deployment solution.
- No specialized software or equipment should be required for this project, beyond a standard software development stations and free IDEs, frameworks, and other tools.

Software and other Deliverables:

- The web application as described above, deployed and tested successfully with real data.
- Must include a complete and clear User Manual for configuring and operating the software.
- A strong as-built report detailing the design and implementation of the product in a complete, clear and professional manner. This document should provide a strong basis for future development of the product.
- Complete professionally-documented codebase, delivered both as a repository in GitHub, BitBucket,or some other version control repository; and as a physical archive on a USB drive
- We require that the created software and GUI be openly available to and accessible by the scientific community (open-source).