Satellite Remote Sensing Analysis Platform

This student team will develop an open-source software platform for analysis of remotely-sensed data. Students on the team will develop all the components of the platform, enabling functionality for a wide variety of users who will apply it to a wide variety of applications.

Project Overview

This student team will develop an open-source software platform for analysis of remotely sensed data. Students on the team will develop all the components of the platform for enabling its use by a wide variety of users who apply it to a wide variety of applications. Such components will consist of the following:

a. C++ library for data storage and processing for satellite imagery, elevation maps, land cover maps, vectors and polygons, 2D and 3D meshes and solid models,
b. Python library and applications for command-line and GUI interaction with the library and the datasets,
c. Web library and applications for image and geographic processing, analysis, and web interactions,
d. Integration of existing high-level analysis codes into the platform,
e. Development of educational modules, including written, web-based, video and multi-media versions,
f. Use of data and metadata standards to provide for interoperability with other systems

Activities in each topic area will be tailored to a range of undergraduate and graduate skill levels and knowledge bases. All the design, planning, decision-making, and implementation activities are performed by the student teams with input and guidance from the faculty member. Students will be expected to develop their expertise throughout / during the team experience.

The project is expected to continue for several years, with improvements, modifications, and added functionality continuing throughout the project duration. An existing open-source framework will be the inspiration for the new design and implementation effort.
Expected Outcomes

In this project, the student team will design, build, test, and use the Satellite Remote Sensing Analysis Platform. The activities will be divided into several sub-teams described in detail below. Students will join a topic-based sub-team, each of which is tasked with developing a use-case analysis, overall architecture, and system design, followed by an implementation plan, testing, and application to a specific science question. We expect that most teams will use a combination of existing open-source libraries and codes as well as custom programming for implementation of the integration framework. All students will participate in developing the design strategy and project planning to bring this endeavor to completion.

Dr. Pierce will contribute guidance at all levels of the project. As students develop skills and seniority within the team we expect that they will grow into leadership roles.

Sub-team 1: C++ Image Processing Library

Students in this team will use existing libraries for data access and storage, while implementing standards for data and metadata representation. A number of open-source image processing libraries will be evaluated and modified as needed for use in this framework. We are targeting high-performance image processing, and so enabling use of parallel resources seamlessly is a major requirement of this portion of the framework. Documentation at several levels will be created simultaneously with the code, including documentation targeted at developers, advanced users, and scientists. Desired skill sets for this team are familiarity with object-oriented programming as well as basic concepts in image processing.

Sub-team 2: C++ Vector Processing Library

Students in this team will use existing libraries for spatial-relational database processing, while implementing existing standards for data and metadata representation and processing. There are many open-source libraries for using vector geographic data, and we will explore their use in this system, modifying and implementing the best-available algorithms and codes. We are targeting high-performance processing, and so enabling the use of parallel resources seamlessly is a major requirement of this portion of the framework. Documentation at several levels will be created simultaneously with the code, including documentation targeted at developers, advanced users, and scientists. Desired skill sets for this team are familiarity with object-oriented programming as well as basic concepts of geometry, geography, and maps.

Sub-team 3: Python Library and GUI

Students in this team will use existing libraries for command-line and GUI development, while implementing existing best practices for code architecture and GUI operation. There are many open-source libraries for creating Python-based command-lines and GUIs, and we will explore their use in this system, modifying and implementing the best-available algorithms and codes. Documentation at several levels will be created simultaneously with the code, including documentation targeted at developers, advanced users, and scientists, including YouTube instructional videos. Desired skill sets for this team are familiarity with object-oriented programming, and event-driven programming, as well as user-computer interaction.

Sub-team 4: Web Library and GUI

Students in this team will use existing libraries for web-based command-line and GUI development, while implementing existing best practices for code architecture and GUI operation. We expect to use a combination of several languages to implement this framework, including JavaScript, PHP, HTML5, and others. There are many open-source libraries for web development, and we will explore their use in this system, modifying and implementing the best-available algorithms and codes. Documentation at several levels will be created simultaneously with the code, including documentation targeted at developers, advanced users, and scientists, including YouTube instructional videos. Desired skill sets for this team are familiarity with object-oriented programming, event-driven programming, and web programming, as well as user-computer interaction.
Preferred Skills for each Subteam

1. C++ Programming
   Preferred Majors:
   EECS, Computer Science, School of Information

   Baseline Qualification:
   Ability and interest in C/C++ and/or Linux applications; this could be obtained in EECS 280: C/C++, Linux

   Expert Qualification:
   EECS 281, EECS 451, C++ data structures

```c++
lua_State *L;
L = luaL_newstate();
luaL_openlibs(L);
if (luaL_loadfile(L, argv[1]) || lua_pcall (L, 0, 1, 0)) {
   lua_close(L);
}
std::vector<double> values = get_array (L, "bodies[2].matrix[3]",2);
for (unsigned int i = 0; i < values.size(); i++) {
   cout << "values" << i << " = " << values[i] << endl;
}
cout << get_string (L, "bodies[1].name") << endl;
cout << get_number (L, "bodies[1].mass") << endl;
cout << get_length (L, "bodies[2].matrix[2]") << endl;
lua_close(L);
return 0;
```
2. Image Processing

Preferred Majors:
EECS, Computer Science, CLaSP, School of Information

Baseline Qualification:
EECS 280: C/C++, Linux

Expert Qualification:
EECS 281 C++ data structures,
EECS 451 digital signal processing,
EECS 442 computer vision,
Math 214 Linear algebra and differential equations,
AOSS 585 Inversion Techniques in Remote Sensing,
NRE 541 Remote Sensing of Environment,
NRE 543 Environmental Spatial Data Analysis
CEE 570 (NRE 569). Introduction to Geostatistics

Books:
Spatial Databases: With Application to GIS, by Philippe Rigaux
Michel O. Scholl, Agnès Voisard

3. Vector Processing

Preferred Majors:
EECS, Computer Science, Civil Engineering, School of Natural Resources

Baseline Qualification:
EECS 280: C/C++, Linux

Expert Qualification:
EECS 281 C++ data structures,
EECS 451 digital signal processing,
NRE 541 Remote Sensing of Environment,
NRE 531 Principles of Geographic Information Systems,
NRE 540 GIS in Natural Resource Applications,
NRE 534 GIS and Landscape Modeling,
NRE 543 Environmental Spatial Data Analysis
CEE 570 (NRE 569). Introduction to Geostatistics
4. Python Programming

Preferred Majors:
EECS, Computer Science, School of Information

Baseline Qualification:
A strong desire to learn and contribute, EECS 280: C/C++, Linux

Expert Qualification:
EECS 281 C++ data structures,
EECS 487 Interactive Computer Graphics,
EECS 493 User Interface Development,
SI 649 Information Visualization,
Coursera Python class: https://www.coursera.org/course/pythonlearn
Books: Learning Python, By Mark Lutz
Programming Python, By Mark Lutz
5. Web Programming

Preferred Majors:
EECS, Computer Science, School of Information

Baseline Qualification:
A strong desire to learn and contribute, EECS 280: C/C++, Linux

Expert Qualification:
EECS 281 C++ data structures,
EECS 487 Interactive Computer Graphics,
EECS 485 Web Database and Information Systems,
EECS 493 User Interface Development,
SI 649 Information Visualization
Udacity javascript course: https://www.udacity.com/course/ud804
Books:  JavaScript: The Definitive Guide, By David Flanagan
        JavaScript Cookbook, By Shelley Powers
        Developing Web Components, By Jarrod Overson, Jason Strimpe
        HTML5 and JavaScript Web Apps, By Wesley Hales
        OpenLayers Cookbook, By Antonio Santiago Perez
        jQuery UI, By Eric Sarrion
        Modern PHP, By Josh Lockhart
        Programming PHP, By Kevin Tatroe, Peter McIntyre, Rasmus Lerdorf
        PHP Cookbook, By David Sklar, Adam Trachtenberg
        PHP Web Services, By Lorna Jane Mitchell
Faculty Sponsor

Leland Pierce

Dr. Leland Pierce is a Research Scientist in the EECS Department who has been involved in remote sensing data processing for the past 25 years. He has been working on the first version of this framework for the past 4 years.

Student Responsibilities

VIP faculty research projects are both a professional and academic learning experience for students. By participating in this program, students are actively preparing for graduate school and a professional career. As part of the experience, MDP expects professional behavior. To best prepare you for future professional opportunities, your experiences on this MDP team will be very broad. In addition to key technical skills that you will bring to the team, you will engage deeply in the self-directed learning of new and important concepts, demonstrate flexibility, collaboration, and cooperation, and develop strong professional communication skills. This also means that you will need to be able to work outside of your traditional area of study in the true multidisciplinary nature of our projects. You won't always be able to anticipate how your skills and expertise will be used, so joining a VIP faculty research team will challenge you to grow and develop as a professional.

Internship

The Multidisciplinary Design Program offers summer research stipends to continue research over the summer. Applications for funding will open in February 2017. Research stipends will not exceed $3,000.

Legal

All students must sign a standard MDP Faculty Research IP Agreement.

The Fine Print

- Application for all projects is open from September 6th, 2016 until 11:59pm on Sunday, October 16th.
- Please schedule an appointment with your home department’s academic advisor before November and make sure you will have time in your schedule to complete a project if offered a position.
- All projects will start in January 2017 and end in December 2017. Summer participation is not required unless otherwise specified in the project description. Project Teams labeled VIP (Faculty Research) will extend for multiple semesters, and long-term participation is highly encouraged.
- Increase your chances of matching to a project by applying for all of the teams that interest you and attend the Project Opportunity Fair on October 5th.
- Successful students will receive one offer to join one project only. Offers will be sent to students starting in November and will continue until all teams are filled (early December).