



Project 009 Geospatially Drive Noise Estimation Module

Georgia Institute of Technology

Project Lead Investigator

Principal Investigator:

Professor Dimitri N. Mavris

Director, Aerospace Systems Design Laboratory

School of Aerospace Engineering

Georgia Institute of Technology

Phone: (404) 894-1557

Fax: (404) 894-6596

Email: dimitri.mavris@ae.gatech.edu

Co-Principal Investigator:

Dr. Holger Pfaender

Aerospace Systems Design Laboratory

School of Aerospace Engineering

Georgia Institute of Technology

Phone: (404) 385-2779

Fax: (404) 894-6596

Email: holger.pfaender@ae.gatech.edu

University Participants

Georgia Institute of Technology

- PIs: Dr. Dimitri Mavris, Dr. Holger Pfaender
- FAA Award Number: 13-C-AJFE-GIT-059
- Period of Performance: June 5, 2020 to June 4, 2021
- Tasks:
 1. Literature Review and GIS Software Evaluation.
 2. Investigation of Emerging Computational Technologies.
 3. Collaboration with UAS Computation Module Development Team.
 4. Noise Computation Engine Integration.

Project Funding Level

This project is funded at the following levels: Georgia Institute of Technology (\$250,000). The Georgia Institute of Technology has agreed to a total of \$250,000 in matching funds. This total includes salaries for the project director, research engineers, and graduate research assistants and for computing, financial, and administrative support, including meeting arrangements. The institute has also agreed to provide tuition remission for students whose tuition is paid via state funds.

Investigation Team

Georgia Institute of Technology

- PI: Dimitri Mavris
- Co-Investigator: Holger Pfaender
- Graduate Students: Aroua Gharbi, Cem Yumuk, Anthony Markowitz

Project Overview

The goal of this task is to develop a novel geospatially driven noise estimation module to support computation of noise resulting from the operation of Unmanned Aircraft Systems (UAS) and other upcoming vehicle concepts. The development

of the module will leverage emerging computational technologies in order to achieve fast and efficient modeling of a potentially large number of vehicles and operations. The module will be designed to be integrated as a component module or plug-in to other applications relying on a Geographic Information System (GIS) interface. The noise estimation approach will be based on the concept of precomputed noise grid tiles addition. The module’s design phase will identify what emerging open-source geospatial and data processing technologies would be best suited to serve as the module’s computational infrastructure and assess if they can provide innovative, maintainable, and affordable solutions.

Task 1 – Literature Review and GIS Software Evaluation

Georgia Institute of Technology

Objectives

Develop a novel geospatially driven noise estimation module to support computation of noise resulting from the operation of UAS and other upcoming vehicle concepts.

Research Approach

This Task aims to identify the leading GIS software. This review focused on open-source options. For an adequate evaluation of the options, six criteria were set forth:

1. Data import: Ability to read shape files format of input geometrical data as well as rasterized (gridded) data.
2. Data storage: Capability to store geospatial data in either shape/vector formats or rasterized data.
3. Geometric calculations: Conversion to and from Cartesian coordinate system and other Earth model coordinates and ability to compute polygon areas and lengths as well as unions and substructions.
4. Geospatial calculations: Ability to perform calculations on given vector or raster data and draw contour plots.
5. Display: Ability to print raw or processed geospatial data as various map displays.
6. Map data: Capability to display results with relation to landmasses, political boundaries such as states and counties, as well as roads and buildings.

QGIS

QGIS is a user-friendly open-source GIS written in C++. It runs on Linux, Unix, Mac OSX, Windows, and Android and supports numerous vectors, raster, and database formats and functionalities. Apart from built-in functionalities, QGIS allows users to install and create their own plugins. New applications can also be created in QGIS through C++ and Python languages.

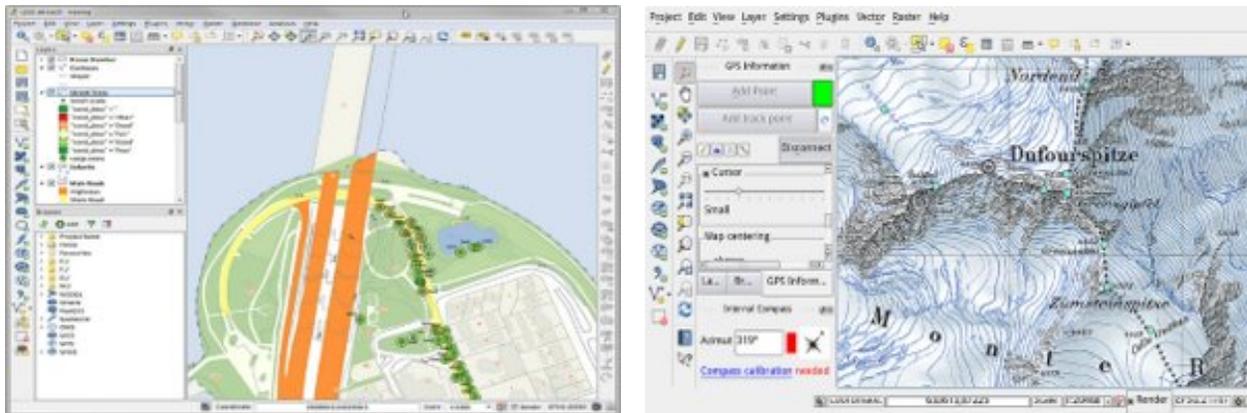


Figure 1. Screenshot of QGIS.

Evaluation Criteria

1. Data import: Imports shape and raster files.
2. Data storage: Stores geospatial data in vector and raster formats.
3. Geometric calculations: Supports Cartesian (x, y), polar (length, angle), projected (x-north, y-east); calculates length or area of geometry features; and provides overlay, union and difference between areas.



4. Geospatial calculations: Creates vector contour map from an elevation raster; and carries out raster to vector conversion.
5. Display: Web mapping is available with QGIS2Web.
6. Map data: Displays geospatial data such as countries, states, and counties as well as roads.

Open JUMP

OpenJUMP is a Java-based open-source GIS. It works on Windows, Linux, and Mac platforms having Java 1.7 or later. Reading and writing vector formats, displaying geospatial data, and execution of geometric calculations are some of OpenJUMP's features. Additional plugins for more capabilities are also available.

Evaluation Criteria

1. Data import: Imports shape and raster files.
2. Data storage: Stores geospatial data in vector and raster formats.
3. Geometric calculations: Supports Coordinate Reference System (CRS), (Cartesian (x, y, z), geographic (longitude, latitude, height) and projected (x-north, y-east)); provides CRS transformation library called PROJ; calculates length or area of geometry features; provides overlay, union and subtraction.
4. Geospatial calculations: Provides conversion between desired file formats (raster to vector conversion); does NOT feature contour plot.
5. Display: Does NOT provide web application.
6. Map data: Display geospatial data such as countries, states, and counties as well as roads.

SAGA (System for Automated Geoscientific Analyses)

SAGA is an open-source cross-platform GIS software written in C++. It can be run on Windows, Linux, FreeBSD, and Mac (OS X). SAGA provides multiple libraries for GIS calculations: digital terrain analysis, image segmentation, fire spreading analysis and simulation, etc. In addition to these libraries, SAGA allows the scripting of custom models through the Command Line Interface and the Python interface.

Evaluation Criteria

1. Data import: Imports shape and raster files.
2. Data storage: Stores geospatial data in vector and raster formats.
3. Geometric calculations: Supports Geographic Coordinate System (latitude, longitude) and Universal Transverse Mercator (UTM).
4. Geospatial calculations: Performs raster to vector conversions and can create contour lines.
5. Display: Displays data with histogram and scatter plot; provides web mapping service.
6. Map data: Enables visualization of spatial data into cartographic maps. It can also import maps from Web Map Service (WMS) and Open Street Map.

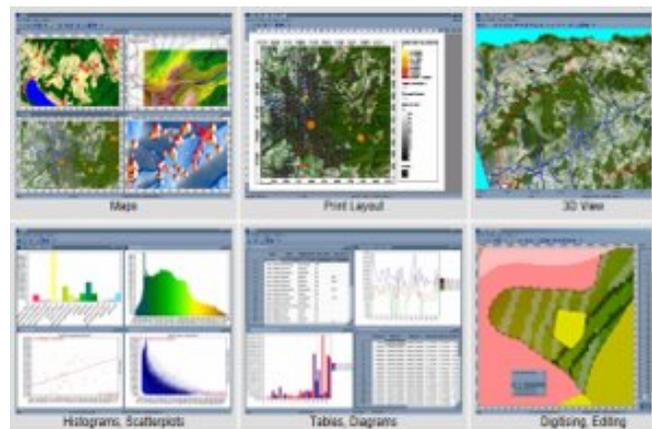
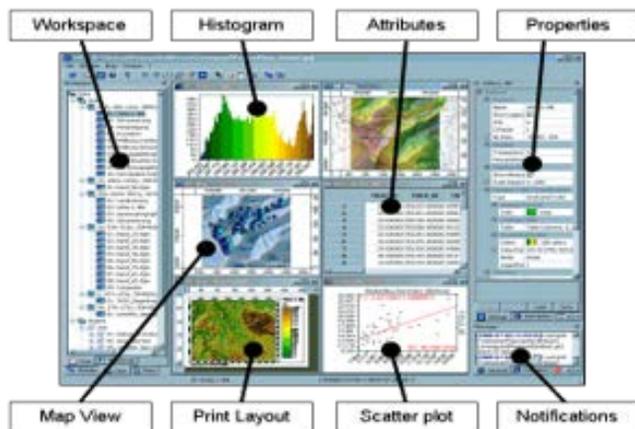


Figure 2. Screenshot of SAGA.



Deck.gl

Deck.gl is a WebGL visualization framework for large data sets. It allows the users to map data (JSON objects, csv) into a stack of layers. These layers can be imported directly from a catalogue or built by the user.

Evaluation Criteria

1. Data import: Reads only csv files.
2. Data storage: Doesn't store geospatial data into vector or shape files.
3. Geometric calculations: Supports Geographic Coordinate System (latitude, longitude) using Web Mercator.
4. Geospatial calculations: Doesn't convert raster data to vector data. Can create contour lines for a given threshold and cell size.
5. Display: Offers an architecture for packaging advanced WebGL based visualizations; enables users to quickly get impressive visual results with limited effort.
6. Map data: Displays geospatial data with relation to roads and buildings easily.

GRASS GIS

GRASS stands for Geographic Resources Analysis Support System. It is an open-source Java-based software for vector and raster geospatial data management, geoprocessing, spatial modelling, and visualization. It has compatibilities with QGIS, meaning that QGIS can run some features of GRASS GIS as a plugin. Already developed addons along with capability to develop own addons are available.

Evaluation Criteria

1. Data import: Imports vector and raster files.
2. Data storage: Stores geospatial data in vector and raster formats.
3. Geometric calculations: Supports Coordinate Reference System (CRS), (Cartesian (x, y, z) and geographic (longitude, latitude, height)); provides CRS transformation library called PROJ; calculates length or area of geometry features; provides overlay, union, and subtraction.
4. Geospatial calculations: Provides conversion between desired file formats (raster to vector conversion); creates contour lines.
5. Display: Provides web mapping service.
6. Map data: Displays geospatial data such as countries and states by using Inkspace.

GeoPandas

GeoPandas is an open-source project developed in Python to provide a useful library for working with geospatial data. It is able to run on distributions of Linux and Windows. It primarily uses the Python packages pandas (as a base for its data storage), shapely (to manipulate the shapes stored in the advanced database), Fiona (for file access), and Descartes and matplotlib (for plotting the visuals of the data). It is most adept at displaying discrete sections of data in a geospatial visualization. It is limited in its ability to display graphics outside of the Python environment and does not support conversion to the desired raster/vector formats.

Evaluation Criteria

1. Data import: Read almost any vector-based spatial data format.
2. Data storage: Store geospatial data in vector and raster formats.
3. Geometric calculations: Supports Coordinate Reference System (CRS); cannot calculate length or area of geometry features; has overlay functions, such as intersections between two or more areas, union (merges the areas of one layer to one single area), difference (A-B areas), and polygons.
4. Geospatial calculations: No conversion to any desired file formats (no raster to vector formats); does not provide contour plot function.
5. Map data: Uses various map projection using CartoPy, Python library.
6. Display: Does not provide web application; good representation in 3D colorspace using matplotlib.



Figure 3. GeoPandas can overlay processed geospatial data over existing maps.

WorldWind

WorldWind is an open-source virtual 3D globe visualization API developed by NASA in partnership with the European Space Agency. It is written in both Java (for desktop and Android devices) and JavaScript (for web applications). After being suspended from development in 2019, it was refunded in August of 2020. It can import a variety of input files with geospatial data, stores the data in both raster and vector formats, does sufficient geometric and geospatial calculations, and produces good visualizations with comprehensive map data.

Evaluation Criteria

1. Data import: Import shapefile, KML, VPF, GML, GeoJSON, GeoRSS, GPX, NMEA, etc.
2. Data storage: Store geospatial data in vector and raster formats.
3. Geometric calculations: Support Geographic Coordinate System (latitude, longitude), Universal Transverse Mercator (UTM), and MGRS; draw and measure distance and area across the terrain.
4. Geospatial calculations: Display contour lines on surface terrain at a specified elevation.
5. Map data: Visual representation of scalar values, like noise, over a grid of geographic positions; visualize the results on web and Android platform.
6. Display: Display geospatial data divided into country, states, and city.

Overall Evaluation

	Intuitive GUI	Compatibility	Statistical Analyses	Data Import	Data Storage	Geometric Calculations	Geospatial Calculations	Map Data	Display	Total
QGIS	3	5	3	5	5	5	5	5	4	40
Open JUMP										
SAGA	3	3	4	5	5	4	5	5	4	38
Deck.gl	4	3	1	1	1	3	3	5	5	26
Kepler.gl										
GRASS										
gvSIG										
MapWindow										
GeoPandas		3		5	5	4	1	2	2	19
WorldWind		5		5	5	4	4	5	5	28

GIS applications can be broadly classified in two categories, desktop and web-based applications.



WebGIS applications use web technologies to display and communicate geospatial information to an end user. There are five common elements in every webGIS application:

1. A web application
 - The interface used by the client. It has tools to visualize, analyze, and interact with geographic information. It can be run on a web browser or a GPS-enabled device.
2. Digital basemaps
 - The geographical context for the application. E.g., transportation, topography, imagery, etc.
3. Operational layers
 - The layer to display the results of an operation. E.g., observations, sensor feeds, query results, analytic results, etc.
4. Tasks and tools in the webGIS application
 - Tools to perform operations beyond mapping.
5. Geodatabase(s)
 - Container of geo data. It can be geodatabase(s), shape files, tabular databases, CAD files, etc.

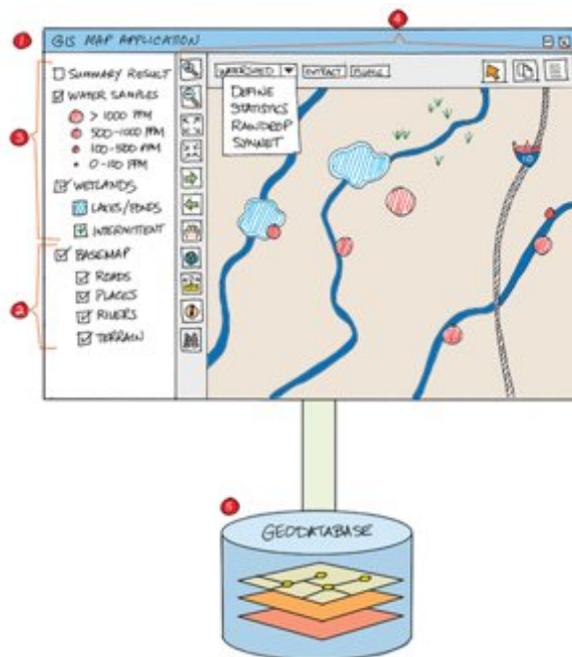


Figure 4. A sketch of a webGIS application.

WebGIS applications come with multiple advantages as well as limitations. These are provided in the non-exhaustive list in table below:

Advantages of WebGIS	Drawbacks of WebGIS
<ul style="list-style-type: none"> • Provide a broader reach for the customers compared to a traditional desktop application • Better cross-platform capability with the different web browsers that can be used • Easy to use for customers with different levels of GIS expertise • Extendable to cloud services hence allowing manipulation and use of big GIS data • Lower cost to entry. Most of the libraries and tools are open source with good community support • Allows real-time analysis 	<ul style="list-style-type: none"> • Harder to build. The developers need to have a good knowledge of multiple scripting languages to build the app (Python, JavaScript, html, etc.) • Security of the data might be dependent on a third party • The application might need to be hosted outside of the organization

The team has also started a dialogue with the Aviation Environmental Design Tool (AEDT) development team as to which GIS functionality will be required in order to be able to integrate the UAS noise engine into AEDT in the future.

Task 2 – Investigation of Emerging Computational Technologies

The Ascent 9 team has started to develop an initial testing plan to investigate the emerging computational technologies with a sample test problem in various GIS and computational environments.

Task 3 – Collaboration with UAS Computation Module Development Team

The ASCENT Project 9 team started an initial dialogue with the UAS Computation Module Development team at Mississippi State University to explore way that both teams can effectively collaborate and exchange data and ideas.

Task 4 – Noise Computation Engine Integration

This task has not been started yet.

Milestone

The team presented an initial overview of candidate GIS systems to FAA and members of the AEDT development team.

Major Accomplishments

None yet.

Publications

None

Outreach Efforts

Initial outreach and coordination with the ASSURE Center of Excellence team and their work at Mississippi State University.

Awards

None

Student Involvement

The Georgia Institute of Technology student team consists of three graduate research assistants (GRA). At the beginning of the project, all seven GRAs engaged in the GIS background research. The team is now being divided into tackling the different aspects and implementation of the noise engine and the novel computational technology testing.

Plans for Next Period

This project is still in the middle of the Year 1 work plan. Therefore, the plan for the reporting period is to finish the current work plan and test the emerging computational technologies on a defined sample problem. After further collaboration with the team at Mississippi State University, the noise computation engine will be integrated into a user interface as planned.