AiGIS2/PyAiGIS, a Python-Based Interactive Analysis and Visualization Tool for Asteroid Exploration Data

Naru Hirata[1, 2], Tsukasa Nagayoshi[2] 1: ARC-Space, University of Aizu, 2: Graduate School of Computer Science and Engineering, University of Aizu

Introduction

AiGIS2/PyAiGIS is a project to develop a code collection of Python or a Python module visualization and analysis on a Jupyter Notebook environment on small body exploration data. This will enable it to work on a Jupyter Notebook or JupyterLab environment, and in combination with Python's versatile modules. it can perform a variety of analyses and visualize the results that were not possible with standalone applications.

Technologies

- Jupyter Notebook/JupyterLab
- Visualization Toolkit (VTK)
- PyVista/PyVistaQt
- Pandas
- SpiceyPy

Development

Two-sided strategy

1.Code sample development (this poster)

Basic procedures of data visualization and analysis are experimentally developed as Python code examples.

These examples are good for

- Early adapters
- Analysts with Python skills

2.Python function and module development

Steps that constitute a typical work procedure are implemented as practical Python functions. These functions and python module are good for

- Beginners
- Non-skilled analysts

Public Release

We share our products through following websites. Any feedback is welcome.

Project website

https://arcspace.jp/aigis2:top

GitHub project site

https://github.com/AiGIS-PyAiGIS/PyAiGIS

Acknowledgments

This work was supported following research funds: MEXT Promotion of Distinctive Joint Research Center Program Grant Number JPMXP0619217839, JPMXP0622717003, JSPS KAKENHI Grant Number 21K03647, and the JAXA Hayabusa2# International Visibility Enhancement Project.

Examples of Basic Features

Importing modules an Data loading

```
[1]: import pyvista as pv
     import pyvistaqt as pvqt
     import vtk
     import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import glob
     import math
     import spiceypy as spice

[2]: obj_low = "./Itokawa_sample/Shapemodel/Itokawa_49152.obj"
     mesh_low = pv.read(obj_low)
```

Visualization

1.Shape model

```
[12]: p = pv.Plotter()
      p.add_mesh(mesh_high, color="w")
      [p.add_mesh(x, line_width=2, color="w") for x in list(lat_grid_high.values())]
      [p.add_mesh(x, line_width=2, color="w") for x in list(lon_grid_high.values())]
      p.show()
```

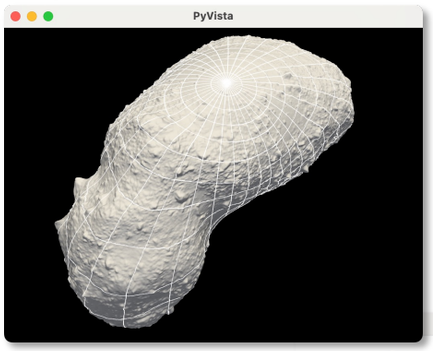
2.Shape model and Mapdata

```
[18]: # p = pv.Plotter()
      p = pv.Plotter(notebook=False)
      p.add_mesh(mesh_low, scalars=mapdata["Surface_Slope"], cmap="coolwarm",
                 scalar_bar_args={'title': 'Surface_Slope [deg]'})
      p.add_scalar_bar(title='Surface_Slope [deg]')
      [p.add_mesh(x, line_width=2, color="w") for x in list(lat_grid.values())]
      [p.add_mesh(x, line_width=2, color="w") for x in list(lon_grid.values())]
      p.show()
```

3.Multiple Views

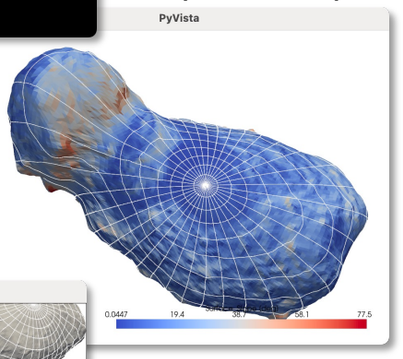
```
[ ]: p0 = pv.Plotter(shape=(2, 2), notebook=False)
     p0.subplot(0, 0)
     p0.add_axes()
     p0.add_text("High-res mesh", font_size=15)
     p0.link_views()
     p0.add_mesh(mesh_high, color='lightgrey',
                 copy_mesh=True)
     [p0.add_mesh(x, line_width=2, color="w") for x in list(lat_grid_high.values())]
     [p0.add_mesh(x, line_width=2, color="w") for x in list(lon_grid_high.values())]

     p0.subplot(0, 1)
     p0.add_axes()
     p0.add_text("Low-res mesh", font_size=15)
     p0.add_mesh(mesh_low, color='lightgrey',
                 copy_mesh=True)
     [p0.add_mesh(x, line_width=2, color="w") for x in list(lat_grid.values())]
     [p0.add_mesh(x, line_width=2, color="w") for x in list(lon_grid.values())]

     p0.subplot(1, 0)
     p0.add_axes()
     p0.add_text("Surface_Slope", font_size=15)
     p0.add_mesh(mesh_low, scalars=mapdata["Surface_Slope"], cmap="coolwarm",
                 scalar_bar_args={'title': 'Surface_Slope [deg]'},
                 copy_mesh=True)
     p0.add_scalar_bar(title='Surface_Slope [deg]')
     [p0.add_mesh(x, line_width=2, color="w") for x in list(lat_grid.values())]
     [p0.add_mesh(x, line_width=2, color="w") for x in list(lon_grid.values())]

     p0.subplot(1, 1)
     p0.add_axes()
```

1.Shape model



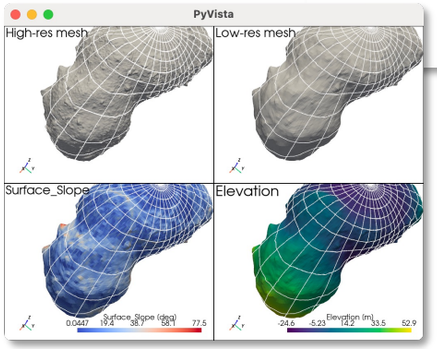
Viewing and illumination angles can be controlled.

2.Shape model and Mapdata



Geographic data (mapdata) can be visualized.

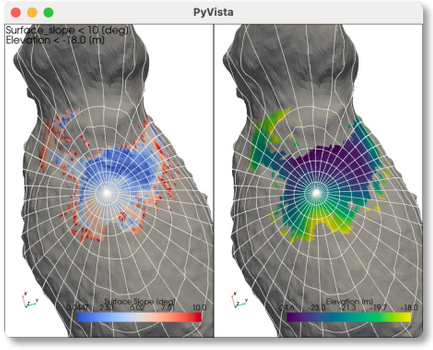
3.Multiple Views



Mouse operation on a view is applied to all views synchronously.

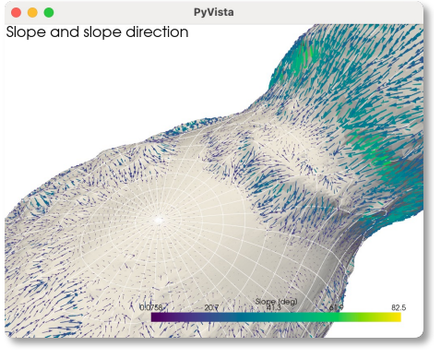
Other Visualization Examples

Conditional color mapping



Only polygons satisfying conditions on the geographic information (slope and elevation) are colorized.

Vector field visualization



Strengths of gravity and downslope directions are visualized by vector arrows.

Other Features

Selection functions: Functions to select a point, line (cross section) and rectangular region by GUI operations. Selections are returned as a list of points (3D coordinates, lat./lon. coordinates, or polygon IDs) for further analysis on the selected area.

Pandas Dataframe-based Analysis: Flexible analysis and visualization workflow becomes possible with a combination of Pandas DataFrame manipulation, plotting and 3D visualization.