

Automated Centerline Extractor

This project allows extracting, processing, and trimming centerlines from vascular models in VTK format. It is designed to automate batch processing and generate results in CSV and/or VTK format.

Requirements

- VTK
- VMTK (Vascular Modeling Toolkit)
- NumPy
- Python 3.8+
- VTK-compatible system (Linux, macOS, Windows)

Installation

Using pip:

```
pip install vtk numpy
```

Using conda (recommended for VMTK):

```
conda install -c vmtk vmtk
```

Command Line Usage

```
python ace.py <input_folder> <output_folder> [pattern] [csv] [vtk] [verbose]
```

Parameters

- `input_folder` → folder containing .vtk files to process.
- `output_folder` → folder where results will be saved.
- `pattern` (optional) → search pattern (default: *_model.vtk).
- `csv` (optional) → True/False, whether to generate CSV files (default: True).
- `vtk` (optional) → True/False, whether to generate VTK files (default: True).
- `verbose` (optional) → True/False, show detailed process information (default: False).

Generated Output

- CSV: list of centerline points with ID, coordinates, and total length.
- VTK: geometry of the trimmed centerlines.

Functions

- Centerline trimming using connectivity, surface normals, local radius, and mesh projection.
- Batch processing of multiple models automatically.

Important Notes

- Higher geometric complexity increases the likelihood of errors and manual extraction needs.
- If a file does not generate valid centerlines, the log will display the corresponding error.
- Visual verification using tools such as Paraview or Slicer 3D is recommended.
- Initial opening detection selects the one with the largest radius.
- Trimming uses multiple verification strategies; slower but more effective.
- Files with suffix _model.vtk are automatically treated as standard input.
- If you improve the code, feedback is appreciated (aguerolucio29@gmail.com).