Scientific Visualization

Half Brain Stucture

Anand Kamble
Department of Scientific Computing
Florida State University

1 Introduction

In this assignment, we are using VTK i.e. The Visualization Toolkit to visualize the given file. We will be using Python as our scripting language.

2 Implementation

The script for this program is written in file `main.py`. In order to run this script, you will need Python and vtk packages installed. I am using Python version 3.10.6 and VTK version 9.3.0. You can install python from https://www.python.org/downloads/ and to install VTK, you will have to run `pip install vtk`

Once everything is installed, you can run the program by running the following command: [1]

Listing 1: bash

```bash
python main.py
```

2.1 Importing the VTK package

To use the VTK package we need to import it into our Python script as follows.

Listing 2: main.py

```python
import vtk
```

2.2 Reading the VTK file

After importing the VTK package, we proceed to read the VTK file in our Python script.

Listing 3: main.py

```python
fran = vtk.vtkPolyDataReader()
fran.SetFileName("Path/To/File/Cort_lobe_poly.vtk")
```

2.3 Data Processing

After reading the VTK file, we apply normal vectors to the polydata using the vtkPolyDataNormals class. This step helps in enhancing the visualization by providing surface orientation information.

Listing 4: main.py

```python
normals = vtk.vtkPolyDataNormals()
normals.SetInputConnection(fran.GetOutputPort())
normals.FlipNormalsOn()
```

[1] You might get a white screen on execution, which means the camera is not looking towards the object, if you rotate your view using mouse cursor you will be able to see the object.

2.4 Creating Visualization

Next, we create a mapper to map the polygonal data into graphics primitives and an actor to represent the mapped data. We set the actor's properties such as color to make the visualization more appealing.

Listing 5: main.py

```python
franMapper = vtk.vtkPolyDataMapper()
franMapper.SetInputConnection(normals.GetOutputPort())
franActor = vtk.vtkActor()
franActor.SetMapper(franMapper)
franActor.GetProperty().SetColor(1.0, 0.49, 0.25)
```

2.5 Setting Up Renderer and Interactor

We set up a renderer and add the actor to it. Additionally, we define a camera to control the viewpoint and clipping range. Finally, we initialize the render window and start the interactor to begin the visualization.

Listing 6: main.py

```python
ren = vtk.vtkRenderer()
renWin = vtk.vtkRenderWindow()
renWin.AddRenderer(ren)

iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renWin)

ren.AddActor(franActor)
ren.SetBackground(1, 1, 1)
renWin.SetSize(800, 800)
renWin.SetWindowName('Cort_lobe_poly.vtk')

cam1 = vtk.vtkCamera()
cam1.SetClippingRange(0.0475572, 2.37786)
cam1.SetFocalPoint(0.052665, -0.129454, -0.0573973)
cam1.SetPosition(0.327637, -0.116299, -0.256418)
cam1.SetViewUp(-0.0225386, 0.999137, 0.034901)
ren.SetActiveCamera(cam1)

iren.Initialize()
renWin.Render()
iren.Start()
```

3 Results

After successfully running the Python script, a window named `Cort_lobe_poly.vtk` should open up and it will look something like this:

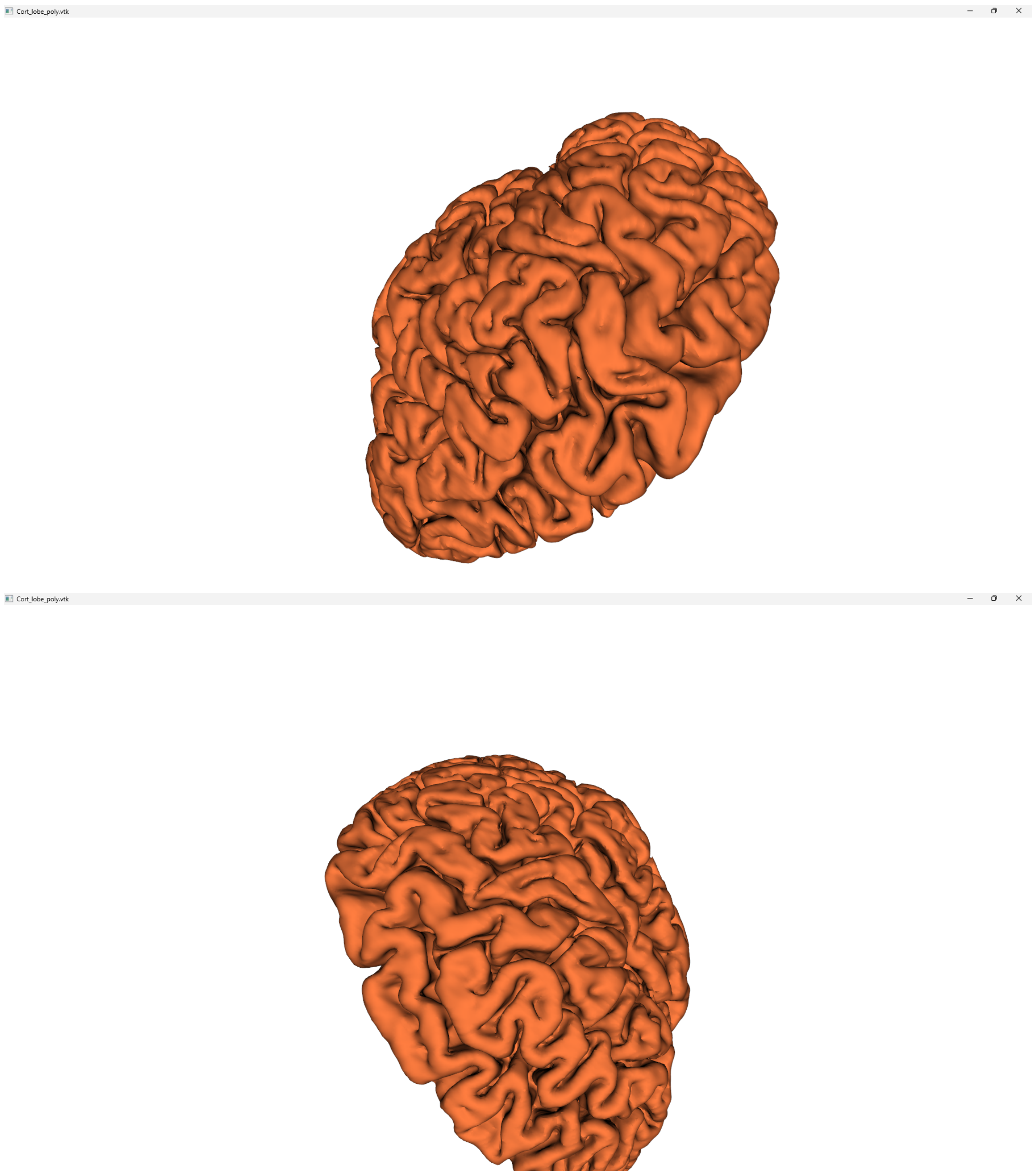

Figure 1: Visualization using VTK