Scientific Visualization

VTK - Isosurfacing

Anand Kamble
Department of Scientific Computing
Florida State University

1 Introduction

The provided code utilizes the Visualization Toolkit (VTK) to visualize medical imaging data from a VTK file named "head.60.vtk". It focuses on extracting and rendering isosurfaces representing skin and bone structures.

2 Implementation

2.1 Data Loading

```
import vtk

# Load the VTK file
filename = "head.60.vtk"
reader = vtk.vtkStructuredPointsReader()
reader.SetFileName(filename)
reader.Update()
```

2.2 Isosurface Extraction

```
# Extract the isosurfaces for bones and skin
iso_skin = vtk.vtkMarchingCubes()
iso_skin.SetInputConnection(reader.GetOutputPort())
iso_skin.SetValue(0, 25)  # Skin value

iso_bone = vtk.vtkMarchingCubes()
iso_bone.SetInputConnection(reader.GetOutputPort())
iso_bone.SetValue(0, 75)  # Bone value
```

2.3 Rendering Setup

```
# Create mapper and actor for skin
skin_mapper = vtk.vtkPolyDataMapper()
skin_mapper.SetInputConnection(iso_skin.GetOutputPort())

skin_actor = vtk.vtkActor()
skin_actor.SetMapper(skin_mapper)
skin_actor.GetProperty().SetColor(1, 0.8, 0.6)  # Skin color

# Create mapper and actor for bones
bone_mapper = vtk.vtkPolyDataMapper()
bone_mapper.SetInputConnection(iso_bone.GetOutputPort())

bone_actor = vtk.vtkActor()
bone_actor.SetMapper(bone_mapper)
bone_actor.GetProperty().SetColor(1, 1, 1)  # Bone color
```

2.4 Rendering Configuration

```
# Create renderer and render window
renderer = vtk.vtkRenderer()
renderer.AddActor(skin_actor)
renderer.AddActor(bone_actor)
renderer.SetBackground(0.1, 0.2, 0.4)

render_window = vtk.vtkRenderWindow()
render_window.SetSize(800, 800)
render_window.AddRenderer(renderer)
```

2.5 Visualization Interaction

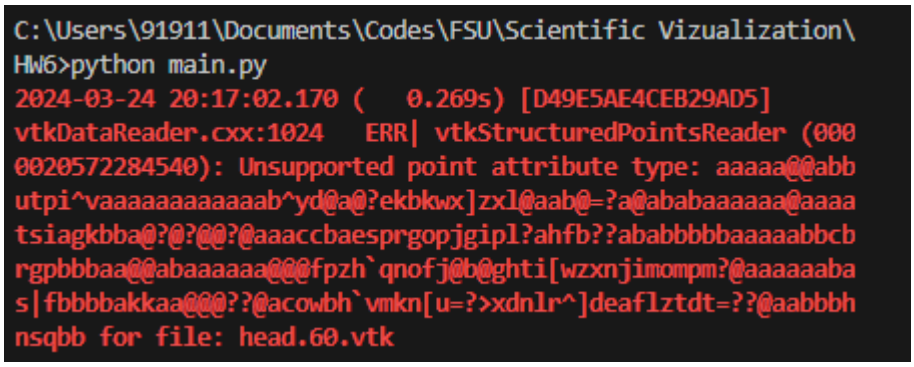
```
# Create render window interactor
interactor = vtk.vtkRenderWindowInteractor()
interactor.SetRenderWindow(render_window)

# Start the visualization
render_window.Render()
interactor.Start()
```

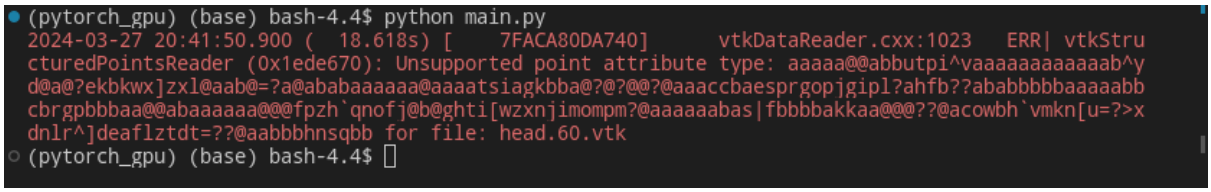
3 Errors

While executing this code, I am getting an error, which I was not able to debug. I have on both Linux and windows, and here are the outputs from both of the machines.

3.1 Windows



3.2 Linux



4 Output

This was the output by VTK.