Graphics and Animation
Assignment 3

Kumar Punithakumar, University of Alberta March 26, 2019

1 Implicit Functions and Cropping

The aim of this assignment is to use a plane to clip a polygonal data, and display the clipped out and remaining parts of the data. You are also expected to show the plane and its intersection with the polygonal data.

1. Read Fuel Oil Heat Exchanger (FOHE) data from file 'fohe.g'
2. Create a plane using `vtkPlane` class. Set the center of the plane to be the center of the FOHE data set. You can use `GetCenter()` function to obtain the center. Set the normal vector to $[0.866, 0.0, -0.5]^T$
3. Clip the data using `vtkClipPolyData` class. Set the clipping value of the implicit function to zero. Use surface and wireframe representation to display the clipped out and remaining parts of the data, respectively.
4. Show the intersection area between the plane and polygonal data. Hint: Use `vtkCutter`, `vtkStripper` and `vtkTriangleFilter` classes.
5. Display the plane. You can use `vtkSampleFunction` and `vtkContourFilter` classes to create a polygonal data from the implicit plane function. Set the bounds of the plane polygonal data to be the same as FOHE data (Hint: Use `GetBounds()` function).

Create four view ports and render different parts of the clipped object as shown below.

View Port 1 Clipped out part (Surface representation)	View Port 2 Remaining part (Wireframe representation)
View Port 3 Intersection area (Surface representation)	View Port 4 Combined clipped out, Remaining and intersection area parts as well as implicit plane

The output display should contain the following components: a) Clipped out part of the FOHE data set as a surface representation (View ports 1 and 4); b) Remaining part of the FOHE data as a wireframe representation (View ports 2 and 4); c) Intersection area between the implicit plane and FOHE data (View ports 3 and 4); and d) The implicit plane that you created in step 2 above (View port 4). Use opacity value 0.2 for rendering the implicit plane.

Synchronize all four view ports by using `GetActiveCamera()` and `SetActiveCamera()` functions of the `vtkRenderer` class so that they all display the same view.

1.1 Submission

You are required to submit the following files for this assignment:

1. A commented code (70%)
2. A JPEG image showing the output (20%)
3. A README file containing details on how to run the code and other information such as VTK version used for writing the code. (10%)

Place your files in a single directory. Zip and submit the file via eClass before April 09, 2019. A penalty of 10% per day will be applied for late submissions.