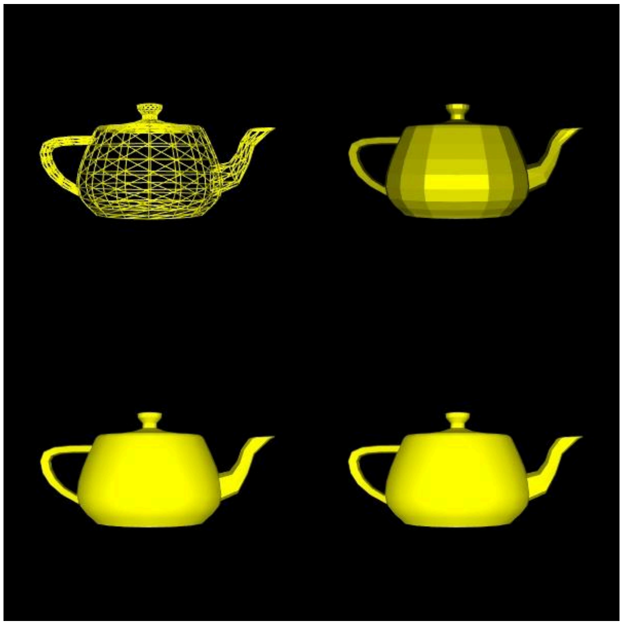
Graphics and Animation

Assignment 1

Model Details

- Name: teapot.stl
- File size: 44 KB
- Number of vertices: 480

Rendered Scene as Image



Source Code Highlights

Full source code available at github: https://github.com/dharanUoA/MM804Assignment1

Reading STL file

Python

```python
reader = vtk.vtkSTLReader()
reader.SetFileName("teapot.stl")
```

Defining mapper

Python

```
mapper = vtk.vtkPolyDataMapper()
mapper.SetInputConnection(reader.GetOutputPort())
```

No shading or texture for actor 1

Python

```
# define actor
actor1 = vtk.vtkActor()
actor1.SetMapper(mapper)
actor1.RotateX(270)

# update properties for representing as wireframe
properties = actor1.GetProperty()
properties.SetColor(1,1,0)
properties.SetRepresentationToWireframe()
```

Define normals

Python

```
normals = vtk.vtkPolyDataNormals()
normals.SetInputConnection(reader.GetOutputPort())
```

Flat shading for actor 2

Python

```
# define new mapper using on normals
mapper = vtk.vtkPolyDataMapper()
mapper.SetInputConnection(normals.GetOutputPort())

# define actor
actor2 = vtk.vtkActor()
actor2.SetMapper(mapper)
actor2.RotateX(270)

# update properties for flat shading
properties = actor2.GetProperty()
properties.SetInterpolationToFlat()
properties.SetColor(1,1,0)
properties.SetDiffuse(0.8)
properties.SetSpecular(0.5)
properties.SetSpecularPower(30.0)
```

Gouraud shading for actor 3

Python

```python
# define new mapper using on normals
mapper = vtk.vtkPolyDataMapper()
mapper.SetInputConnection(normals.GetOutputPort())

# define actor
actor3 = vtk.vtkActor()
actor3.SetMapper(mapper)
actor3.RotateX(270)

# update properties for gouraud shading
properties = actor3.GetProperty()
properties.SetInterpolationToGouraud()
properties.SetColor(1,1,0)
properties.SetDiffuse(0.8)
properties.SetAmbient(0.3)
properties.SetSpecular(0.5)
properties.SetSpecularPower(30.0)
```

Phong shading for actor 4

Python

```python
# define new mapper using on normals
mapper = vtk.vtkPolyDataMapper()
mapper.SetInputConnection(normals.GetOutputPort())

# define actor
actor4 = vtk.vtkActor()
actor4.SetMapper(mapper)
actor4.RotateX(270)

# update properties for phong shading
properties = actor4.GetProperty()
properties.SetInterpolationToPhong()
properties.SetColor(1, 1, 0)
properties.SetDiffuse(0.8)
properties.SetAmbient(0.3)
properties.SetSpecular(0.5)
properties.SetSpecularPower(30.0)
```

Divide the window in 4 renderers and add actors to each of them

Python

```
renderer1 = vtk.vtkRenderer()
renderer1.SetViewport(0, 0.5, 0.5, 1) # top left
renderer1.AddActor(actor1)

renderer2 = vtk.vtkRenderer()
renderer2.SetViewport(0.5, 0.5, 1, 1) # top right
renderer2.AddActor(actor2)

renderer3 = vtk.vtkRenderer()
renderer3.SetViewport(0, 0, 0.5, 0.5) # bottom left
renderer3.AddActor(actor3)

renderer4 = vtk.vtkRenderer()
renderer4.SetViewport(0.5, 0, 1, 0.5) # bottom right
renderer4.AddActor(actor4)
```

Add renderers to render window

Python

```
renWin = vtk.vtkRenderWindow()
renWin.SetSize(600, 600)
renWin.AddRenderer(renderer1)
renWin.AddRenderer(renderer2)
renWin.AddRenderer(renderer3)
renWin.AddRenderer(renderer4)
renWin.Render()
```

Output the image or create window interactor based on input

Python

```
output_to_image = True

if output_to_image:
  # write to image file
  window_to_image_filter = vtk.vtkWindowToImageFilter()
  window_to_image_filter.SetInput(renWin)
  window_to_image_filter.SetInputBufferTypeToRGB()
  window_to_image_filter.ReadFrontBufferOff()
  window_to_image_filter.Update()

  jpg_writer = vtk.vtkJPEGWriter()
  jpg_writer.SetFileName("output_image.jpg")
  jpg_writer.SetInputConnection(window_to_image_filter.GetOutputPort())
  jpg_writer.Write()
else:
  # create interaction window
  iren = vtk.vtkRenderWindowInteractor()
  iren.SetInteractorStyle(vtk.vtkInteractorStyleTrackballActor())
  iren.SetRenderWindow(renWin)
  iren.Initialize()
  iren.Start()
```

Readme.md

About The Assignment

The aim of this assignment is to create multiple view ports to compare different types of representation of objects, namely, wireframe, and surface with flat, Gouraud and Phong shading.

Built With

- Python 3
- vtk 9.3.0

Getting Started

Prerequisites

Download and install python 3 from https://www.python.org/.

Installation (using command prompt)

Unset

```
# Install vtk package
pip install vtk
```

Usage

Run (using command prompt)

Unset

```
python main.py
```

Configuration

Set `output_to_image` variable value.

- `True`: for exporting the rendered scene as Image. The exported image will be stored in the root folder same as main.py named as `output_image.jpg`.
- `False`: for interacting with the render scene.