

## MM - 804 Assignment 2

### Github Repository Link:

[https://github.com/SaiBhargav30393/MM804\\_Assignment2](https://github.com/SaiBhargav30393/MM804_Assignment2)

### Requirements

vtk==9.1.0

python==3.8.8

### Image description

Name - Warlock\_repaired.stl

File size – 31,330 kb

Number of vertices on the original object = 319564

Number of vertices on the remaining part = 115705

Number of vertices on the clipped part = 209519

Number of vertices on the sliced area = 2830

### Source Code:

```
import vtk
from vtk.util.colors import brown_ochre, tomato, banana

# Read 3d model input using vtkSTLReader
statue = vtk.vtkSTLReader()
statue.SetFileName("Warlock_repaired.stl")
# creating a mapper for the stl 3d model
mapper_statue = vtk.vtkPolyDataMapper()
mapper_statue.SetInputConnection(statue.GetOutputPort())
# get normals of the model
statue_center = mapper_statue.GetCenter()
normal_values = vtk.vtkPolyDataNormals()
normal_values.SetInputConnection(statue.GetOutputPort())
# create a plane for the model
statue_plane = vtk.vtkPlane()
statue_plane.SetOrigin(statue_center)
statue_plane.SetNormal(1,0,1)
# create a clipper to clip the 3d model
model_clipper = vtk.vtkClipPolyData()
model_clipper.SetInputConnection(normal_values.GetOutputPort())
model_clipper.SetClipFunction(statue_plane) #Plane to clip
model_clipper.GenerateClipScalarsOn() #present scalar values of the 3d model
model_clipper.GenerateClippedOutputOn() # get clipped data
model_clipper.SetValue(0) # clipping values to 0.
# Map the clipped data
model_clip_map = vtk.vtkPolyDataMapper()
model_clip_map.SetInputConnection(model_clipper.GetOutputPort())
model_clip_map.ScalarVisibilityOff()
```

```

set_background_values = vtk.vtkProperty()
set_background_values.SetDiffuseColor(tomato)

# clip actor
statue_clip_actor = vtk.vtkActor()
statue_clip_actor.SetMapper(model_clip_map)
statue_clip_actor.GetProperty().SetColor(brown_ochre) # Set clipped data
colour
statue_clip_actor.SetBackfaceProperty(set_background_values)

# 4. intersection area between the plane and polygonal.
model_edge_cuts = vtk.vtkCutter()
model_edge_cuts.SetInputConnection(normal_values.GetOutputPort())
model_edge_cuts.SetCutFunction(statue_plane)
model_edge_cuts.GenerateCutScalarsOn()
model_edge_cuts.SetValue(0, 0)
#plane cut
model_strips_cut = vtk.vtkStripper()
model_strips_cut.SetInputConnection(model_edge_cuts.GetOutputPort())
model_strips_cut.Update()
#polygon cut
model_poly_cut = vtk.vtkPolyData()
model_poly_cut.SetPoints(model_strips_cut.GetOutput().GetPoints())
model_poly_cut.SetPolys(model_strips_cut.GetOutput().GetLines()) # Display
the data

# triangle Filter
model_triangle_cut = vtk.vtkTriangleFilter()
model_triangle_cut.SetInputData(model_poly_cut)
model_mapper_cut = vtk.vtkPolyDataMapper()
model_mapper_cut.SetInputData(model_poly_cut)
model_mapper_cut.SetInputConnection(model_triangle_cut.GetOutputPort())
model_actor_cut = vtk.vtkActor()
model_actor_cut.SetMapper(model_mapper_cut)
model_actor_cut.GetProperty().SetColor(banana)

# mapper and actor for remaining data
model_remaining_mapper = vtk.vtkPolyDataMapper()
model_remaining_mapper.SetInputData(model_clipper.GetClippedOutput())
model_remaining_mapper.ScalarVisibilityOff()
model_remaining_actor = vtk.vtkActor()
model_remaining_actor.SetMapper(model_remaining_mapper)
model_remaining_actor.GetProperty().SetRepresentationToWireframe()

# Initialize render window
display_window = vtk.vtkRenderWindow()
display_window.SetSize(1300, 950) #Seting window size of the output rendering
model

```

```

model_window_interactor = vtk.vtkRenderWindowInteractor()
model_window_interactor.SetRenderWindow(display_window)

# Port co-ordinates
min_values_x=[0,.5,0,.5]
max_values_x=[0.5,1,0.5,1]
min_values_y=[0,0,.5,.5]
max_values_y=[0.5,0.5,1,1]

# setting up ports and location
# Port 3
model_display_view3 = vtk.vtkRenderer()
display_window.AddRenderer(model_display_view3)
model_display_view3.SetViewport(min_values_x[0],min_values_y[0],max_values_x[0],max_values_y[0])
# Port 4
model_display_view4 = vtk.vtkRenderer()
display_window.AddRenderer(model_display_view4)
model_display_view4.SetViewport(min_values_x[1],min_values_y[1],max_values_x[1],max_values_y[1])
# port 1
model_display_view1 = vtk.vtkRenderer()
display_window.AddRenderer(model_display_view1)
model_display_view1.SetViewport(min_values_x[2],min_values_y[2],max_values_x[2],max_values_y[2])
# Port 2
model_display_view2 = vtk.vtkRenderer()
display_window.AddRenderer(model_display_view2)
model_display_view2.SetViewport(min_values_x[3],min_values_y[3],max_values_x[3],max_values_y[3])
# Add actors to viewports
model_display_view1.AddActor(stature_clip_actor)
model_display_view3.AddActor(model_actor_cut)
model_display_view2.AddActor(model_remaining_actor)
model_display_view4.AddActor(stature_clip_actor)
model_display_view4.AddActor(model_actor_cut)
model_display_view4.AddActor(model_remaining_actor)
# SetActiveCameras to the ActiveCamera of the first renderer
model_display_view2.SetActiveCamera(model_display_view1.GetActiveCamera());
model_display_view4.SetActiveCamera(model_display_view1.GetActiveCamera());
model_display_view3.SetActiveCamera(model_display_view1.GetActiveCamera());
model_display_view1.ResetCamera()
display_window.Render()
display_window.SetWindowName('MM804 - Graphics and Animation - Assignment 2')
to_image = vtk.vtkWindowToImageFilter()
to_image.SetInput(display_window)
to_image.ReadFrontBufferOff()
to_image.Update()

```

```
write_image = vtk.vtkJPEGWriter() #jpeg file writer to display output to an
image
write_image.SetFileName('output.jpg') # initial camera view output jpeg
filename.
write_image.SetInputConnection(to_image.GetOutputPort()) # Displaying the
rendering scene
write_image.Write()
model_window_interactor.Start()
```

**Output :**





