MM 804 - Graphics and Animations

Link to GitHub:

Downloaded the dataset from

https://nbia.cancerimagingarchive.net/nbia-search/?MinNumberOfStudiesCriteria=1&CollectionCriteria=Lung-PET-CT-Dx

Information about medical dataset used

Name: Lung-PET-CT-Dx

Dimension: 29.1 MB or 29,009,518 bytes

Number of Images: 55

Voxel Resolution: 5mm

Minimum & Maximum pixel intensities: 0 to 255 (Images are 8-bit gray scale)

VTK version

vtk==9.1.0

Setup on Mac and Windows

1. Install Python3 version 3.8.8

   Refer the webiste https://docs.python-guide.org/starting/install3/osx/)

2. Install the requirements package

   pip install -r requirements.txt

3. Run the Script

   python assignment.py

Code

```
import vtk
vtk_reader = vtk.vtkDICOMImageReader() # vtkDICOMImageReader() is used to read the
dataset.
vtk_reader.SetDirectoryName("Lung")
vtk_reader.Update()

vtk_color_Transfer = vtk.vtkColorTransferFunction() # vtkColorTransferFunction() is
used to colour transfer function using the following values.
vtk_color_Transfer.AddRGBPoint(-3024, 0.0, 0.0, 0.0)
vtk_color_Transfer.AddRGBPoint(-77, 0.5, 0.2, 0.1)
vtk_color_Transfer.AddRGBPoint(94, 0.9, 0.6, 0.3)
vtk_color_Transfer.AddRGBPoint(179, 1.0, 0.9, 0.9)
vtk_color_Transfer.AddRGBPoint(260, 0.6, 0.0, 0.0)
vtk_color_Transfer.AddRGBPoint(3071, 0.8, 0.7, 1.0)


vtk_opacity_Transfer = vtk.vtkPiecewiseFunction() # vtkPiecewiseFunction() used to
opacity transfer function using the following values.
vtk_opacity_Transfer.AddPoint(-3024, 0.0)
vtk_opacity_Transfer.AddPoint(-77, 0.0)
vtk_opacity_Transfer.AddPoint(180, 0.2)
vtk_opacity_Transfer.AddPoint(260, 0.4)
vtk_opacity_Transfer.AddPoint(3071, 0.8)


vtk_ct_Mapper = vtk.vtkSmartVolumeMapper() # Create viewports as shown below and
render the CT dataset
vtk_ct_Mapper.SetInputConnection(vtk_reader.GetOutputPort()) # using direct volume
rendering approach in viewport 1.


vtk_ct_Prop = vtk.vtkVolumeProperty() # Opacity and colour transfer functions are
defined already
vtk_ct_Prop.SetScalarOpacity(vtk_opacity_Transfer)
vtk_ct_Prop.SetColor(vtk_color_Transfer)
vtk_ct_Prop.ShadeOn()
```

```
vtk_ct_Volume = vtk.vtkVolume() # Volume actor is defined
vtk_volRen = vtk.vtkRenderer()
vtk_ct_Volume.SetMapper(vtk_ct_Mapper) # Setting the volume actor properties
vtk_ct_Volume.SetProperty(vtk_ct_Prop)

vtk_volRen.AddVolume(vtk_ct_Volume)


vtk_iso = vtk.vtkMarchingCubes() # In viewport 2, display the iso-surface extracted at
intensity
vtk_iso.SetInputConnection(vtk_reader.GetOutputPort()) # value 300 using marching
cubes algorithm.
vtk_iso.ComputeGradientsOn()
vtk_iso.ComputeScalarsOff()
vtk_iso.SetValue(0, 300)

vtk_iso_Mapper = vtk.vtkPolyDataMapper() # Polydata mapper for the iso-surface is
created.
vtk_iso_Mapper.SetInputConnection(vtk_iso.GetOutputPort())
vtk_iso_Mapper.ScalarVisibilityOff()

vtk_iso_Actor = vtk.vtkActor() # Actor for the iso surface
vtk_iso_Actor.SetMapper(vtk_iso_Mapper)
vtk_iso_Actor.GetProperty().SetColor(1.,1.,1.)

vtk_iso_Ren = vtk.vtkRenderer()
vtk_iso_Ren.AddActor(vtk_iso_Actor) # add the actors to the renderer

vtk_combo_Ren = vtk.vtkRenderer() # Created a combo rederer
vtk_combo_Ren.AddActor(vtk_iso_Actor) # Reuse actor and volume
vtk_combo_Ren.AddVolume(vtk_ct_Volume)

x_mins=[0,0.33,0.66] # Create a render window with three viewports
x_maxs=[0.33,0.66,1]
y_mins=[0,0,0]
y_maxs=[1,1,1]

mainWindow = vtk.vtkRenderWindow()
windInteract = vtk.vtkRenderWindowInteractor()
mainWindow.SetSize(1300,600)
windInteract.SetRenderWindow(mainWindow)


vtk_iso_Ren.SetActiveCamera(vtk_volRen.GetActiveCamera()) # SetActiveCameras to the
ActiveCamera of the first renderer
vtk_combo_Ren.SetActiveCamera(vtk_iso_Ren.GetActiveCamera())
vtk_volRen.ResetCamera()
```

```
# Add the renderes to main window
mainWindow.AddRenderer(vtk_volRen)
mainWindow.AddRenderer(vtk_iso_Ren)
mainWindow.AddRenderer(vtk_combo_Ren)

# Set the location
vtk_volRen.SetViewport(x_mins[0],y_mins[0],x_maxs[0],y_maxs[0])
vtk_iso_Ren.SetViewport(x_mins[1],y_mins[1],x_maxs[1],y_maxs[1])
vtk_combo_Ren.SetViewport(x_mins[2],y_mins[2],x_maxs[2],y_maxs[2])
mainWindow.Render()
wind2Im = vtk.vtkWindowToImageFilter()
wind2Im.SetInput(mainWindow)
wind2Im.Update()
writer = vtk.vtkJPEGWriter() # Save the output
writer.SetInputConnection(wind2Im.GetOutputPort())
writer.SetFileName('output_final.jpg')
writer.Write()
windInteract.Initialize()
windInteract.Start()
```

OutPut (From all different angles)