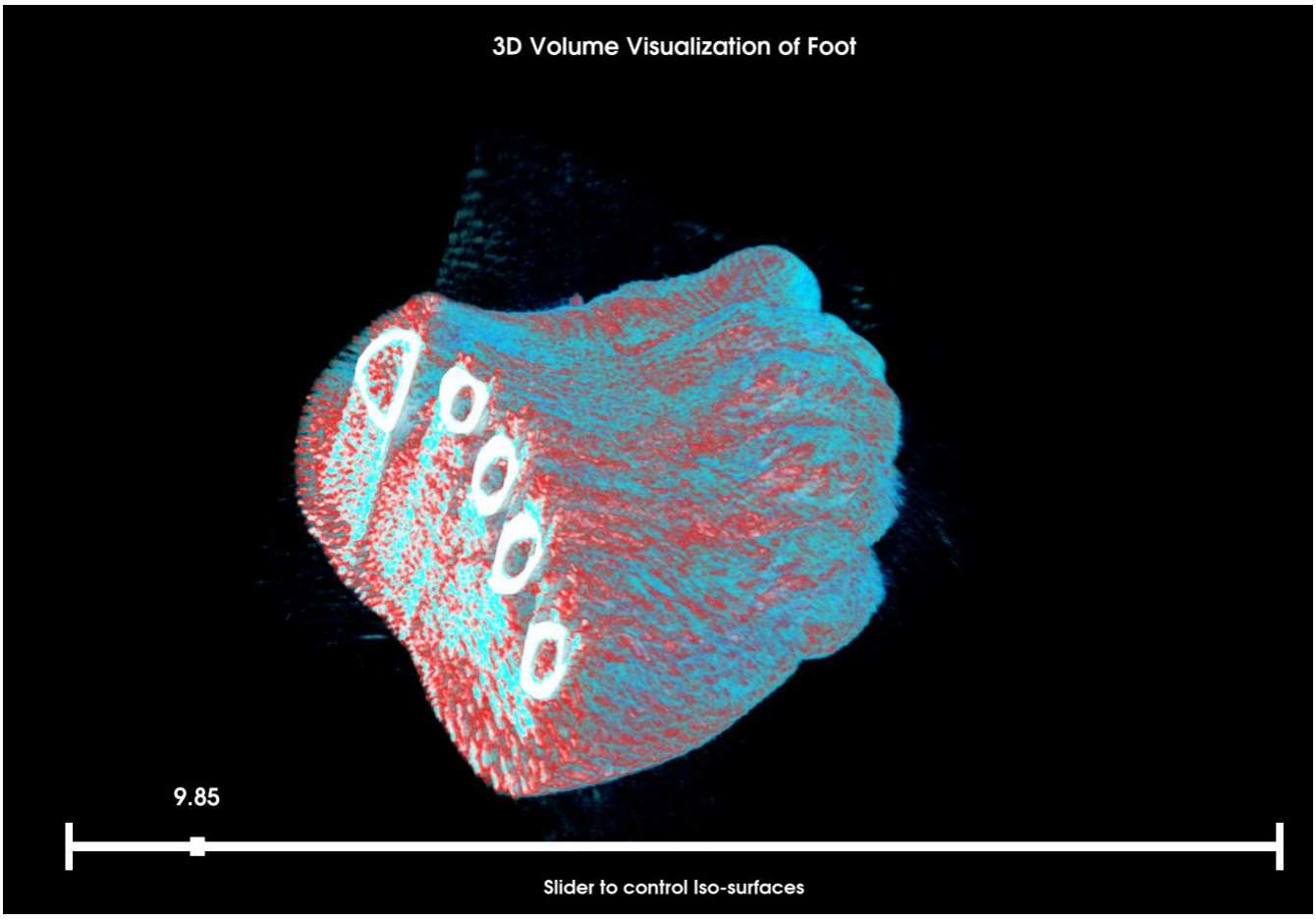
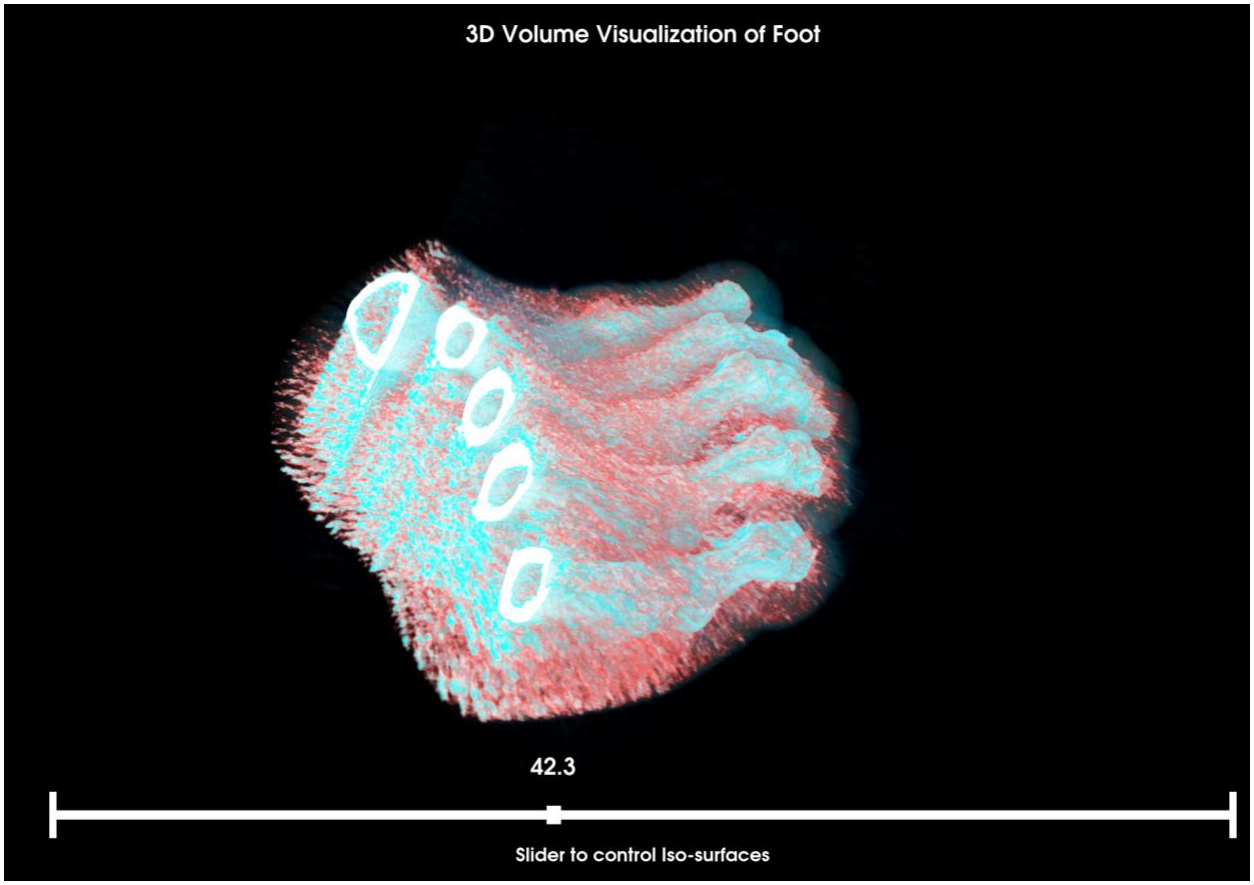
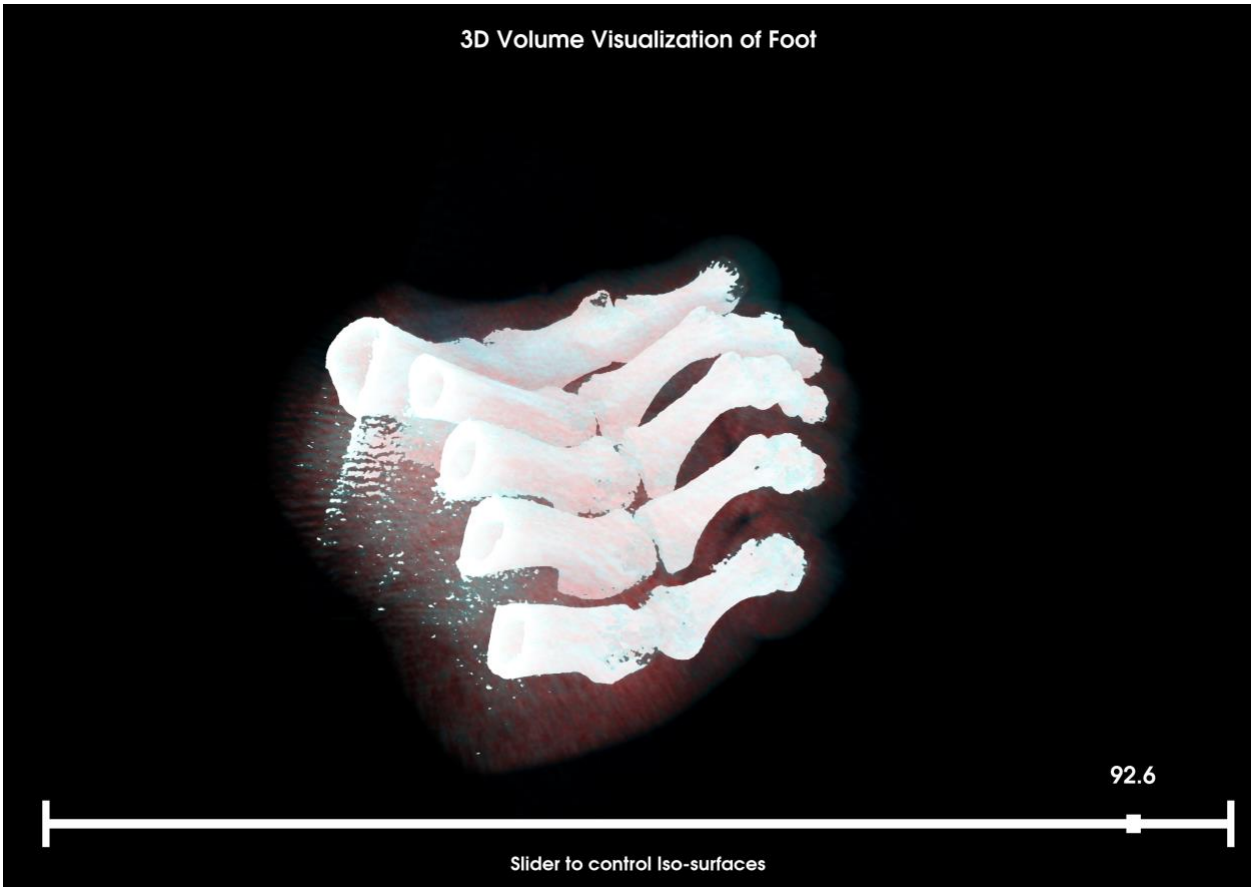
CSCE 679 HW4 Scientific Visualization
UIN: 232009024
Name: Ashutosh Chauhan

Dataset Used: 3D volume dataset (foot): [vtk-js-datasets/data/vti/foot.vti at master · Kitware/vtk-js-datasets (github.com)Links to an external site.]

3D Model Images:




3D Volume Visualization of Foot
42.3
Slider to control Iso-surfaces


3D Volume Visualization of Foot
92.6
Slider to control Iso-surfaces

A description about the underlying data and an explanation as to how you chose to visualize it (e.g., how did you determine suitable isosurface values?)

foot.vti dataset that I used in the visualization contains x, y and z coordinates for all the point along with an associated intensity value. I have used a color transfer function to allocate unique color values to different intensity. I have used blue color for intensity 0 and white color to signify intensity 100. I have used other colors (red, green, cyan, etc) for in-between values. Additionally, I am using an opacity transfer function to add transparency so that multiple isosurfaces can be seen at the same time. I have added a slider that is only showing all the intensity value above the selected position. This is helping is to easily visualize different isosurfaces easily. I have also added different interactions and controls to the plot that is helping me to zoom and rotate the 3D plot as desired.

```
vtkXMLImageDataReader (0x147653050)
  Debug: Off
  Modified Time: 91
  Reference Count: 2
  Registered Events: (none)
  Executive: 0x13771d760
  ErrorCode: Undefined error: 0
  Information: 0x14764e150
  AbortExecute: Off
  Progress: 1
  Progress Text: (None)
  FileName: foot.vti
  CellDataArraySelection: 0x1476540a0
  PointDataArraySelection: 0x14760c7c0
  ColumnArraySelection: 0x14760c7c0
  TimeDataStringArray: 0x1476541a0
  Stream: (none)
  TimeStep:0
  ActiveTimeDataArrayName:TimeValue
  NumberOfTimeSteps:0
  TimeStepRange:(0,0)
  WholeSlices: 1
```

```
vtkVolume (0x1264a5800)
  Debug: Off
  Modified Time: 590964
  Reference Count: 2
  Registered Events: (none)
  Dragable: On
  Pickable: On
  AllocatedRenderTime: 10
  EstimatedRenderTime: 0
  NumberOfConsumers: 1
  RenderTimeMultiplier: 1
  Visibility: On
  PropertyKeys: none.
  useBounds: 1
  IsIdentity: true
  Position: (0, 0, 0)
  Orientation: (0, 0, 0)
  Origin: (0, 0, 0)
  Scale: (1, 1, 1)
  Bounds:
    Xmin,Xmax: (0, 255)
    Ymin,Ymax: (0, 255)
    Zmin,Zmax: (0, 255)
  UserTransform: (none)
  UserMatrix: (none)
  Property:
    Debug: Off
    Modified Time: 590963
    Reference Count: 2
    Registered Events: (none)
    Independent Components: On
    Interpolation Type: Linear
    Use Clipped Voxel Intensity: Off
    Clipped Voxel Intensity: -1e+38
    Properties for material 0
    Color Channels: 3
    RGB Color Transfer Function: 0x117370e10
    Scalar Opacity Transfer Function: 0x117370f70
    Gradient Opacity Transfer Function: 0x0
    DisableGradientOpacity: Off
    2D Transfer Function: 0x0
    ComponentWeight: 1
    Shade: 0
        Ambient: 0.1
        Diffuse: 0.7
        Specular: 0.2
        SpecularPower: 10
```

```
vtkOpenGLRenderer (0x15aaacc00)
  Debug: Off
  Modified Time: 590965
  Reference Count: 1
  Registered Events: (none)
  Aspect: (1, 1)
  PixelAspect: (1, 1)
  Background: (0, 0, 0)
  Background2: (0.2, 0.2, 0.2)
  BackgroundAlpha: 0
  GradientBackground: Off
  GradientMode: VTK_GRADIENT_VERTICAL
  Viewport: (0, 0, 1, 1)
  Displaypoint: (0, 0, 0)
  Viewpoint: (0, 0, 0)
  Worldpoint: (0, 0, 0, 0)
  Pick Position X1 Y1: -1 -1
  Pick Position X2 Y2: -1 -1
  PickedZ: 1
  Props:
    Debug: Off
    Modified Time: 591207
    Reference Count: 1
    Registered Events: (none)
    Number Of Items: 2
  PickResultProps:
  nullptr
  Near Clipping Plane Tolerance: 0
  ClippingRangeExpansion: 0.5
  Ambient: (1, 1, 1)
  Backing Store: Off
  Display Point: (0, 0, 0)
  Lights:
    Debug: Off
    Modified Time: 590968
    Reference Count: 1
    Registered Events: (none)
    Number Of Items: 0
  Light Follow Camera: On
  View Point: (0, 0, 0)
  Two Sided Lighting: On
  Automatic Light Creation: On
  Layer = 0
  PreserveDepthBuffer: Off
  PreserveColorBuffer: Off
  Interactive = On
  Allocated Render Time: 100
  Last Time To Render (Seconds): -1
  TimeFactor: 1
  Erase: On
  Draw: On
  UseDepthPeeling: Off
  OcclusionRation: 0
  MaximumNumberOfPeels: 4
  LastRenderingUsedDepthPeeling: Off
  Delegate:null
```

If you include interactions or animations for additional points, link to a video (preferably hosted on YouTube or Google Drive) showcasing these.

https://youtu.be/NOYSHrnvrfQ

Your entire code at the end of the report. Code should include high level comments such as "loading dataset" or "rendering visualization" so that the goal of each section is clear.

```python
import vtk
# Loading the 3D foot dataset
reader = vtk.vtkXMLImageDataReader()
# reader.SetFileName('mri_ventricles.vti')
reader.SetFileName('foot.vti')
reader.Update()

# Create a volume mapper and volume property
mapper = vtk.vtkSmartVolumeMapper()
property = vtk.vtkVolumeProperty()
property.ShadeOff()
mapper.SetInputConnection(reader.GetOutputPort())
property.SetInterpolationTypeToLinear()

# Define a color and opacity transfer function and adding differnet color/opacity points to plot
colorTF = vtk.vtkColorTransferFunction()
opacityTF = vtk.vtkPiecewiseFunction()
colorTF.AddRGBPoint(0, 0, 0, 1)
opacityTF.AddPoint(0, 0)
colorTF.AddRGBPoint(20, 0, 1, 1)
opacityTF.AddPoint(20, 0.02)
colorTF.AddRGBPoint(40, 1, 0, 0)
opacityTF.AddPoint(40, 0.02)
colorTF.AddRGBPoint(60, 1, 1, 1)
opacityTF.AddPoint(60, 0.1)
colorTF.AddRGBPoint(80, 0, 1, 1)
opacityTF.AddPoint(80, 0.2)
colorTF.AddRGBPoint(100, 1, 1, 1)
opacityTF.AddPoint(100, 1)
property.SetColor(colorTF)
property.SetScalarOpacity(opacityTF)

# Creating a volume and renderer and adding volume to the renderer
volume = vtk.vtkVolume()
volume.SetMapper(mapper)
volume.SetProperty(property)
renderer = vtk.vtkRenderer()
renderer.AddVolume(volume)
```

```
# Creating an actor to add heading of plot
heading = vtk.vtkOpenGLTextActor()
heading.SetTextScaleModeToNone()
heading.GetPositionCoordinate().SetCoordinateSystemToNormalizedDisplay()
heading.SetPosition(0.5, 0.95)
heading.GetTextProperty().SetFontSize(45)
heading.GetTextProperty().SetBold(1)
heading.GetTextProperty().SetShadow(1)
heading.GetTextProperty().SetColor(1.0, 1.0, 1.0)
heading.GetTextProperty().SetJustificationToCentered()
heading.GetTextProperty().SetVerticalJustificationToTop()
heading.SetInput("3D Volume Visualization of Foot")
renderer.AddActor(heading)

# Creating a render window and adding interaction to it
renderWindow = vtk.vtkRenderWindow()
renderWindow.SetWindowName("3D Volume Visualization of Foot")
renderWindow.SetSize(1200, 900)
renderWindow.AddRenderer(renderer)
renderWindowInterface = vtk.vtkRenderWindowInteractor()
renderWindowInterface.SetRenderWindow(renderWindow)
interactor_style = vtk.vtkInteractorStyleTrackballCamera()
interactor_style.SetMouseWheelMotionFactor(1.0)
renderWindowInterface.SetInteractorStyle(interactor_style)
```

```
# Adding a slider widget in plot to control iso-surfaces
slider = vtk.vtkSliderRepresentation2D()
slider.SetMinimumValue(0)
slider.SetMaximumValue(100)
slider.SetValue(45)
slider.GetPoint1Coordinate().SetCoordinateSystemToNormalizedDisplay()
slider.GetPoint1Coordinate().SetValue(0.1, 0.1)
slider.GetPoint2Coordinate().SetCoordinateSystemToNormalizedDisplay()
slider.GetPoint2Coordinate().SetValue(0.9, 0.1)
slider.SetTitleText("Slider to control Iso-surfaces")
slider.SetTitleHeight(0.02)
slider_widget = vtk.vtkSliderWidget()
slider_widget.SetInteractor(renderWindowInterface)
slider_widget.SetRepresentation(slider)
slider_widget.SetAnimationModeToAnimate()
slider_widget.EnabledOn()

# Function to update opacity transfer function based on slider value
def opacity_control_via_slider(widget, event):
    value = widget.GetRepresentation().GetValue()
    opacityTF.RemoveAllPoints()
    opacityTF.AddPoint(0, 0)
    opacityTF.AddPoint(value, 0.02)
    opacityTF.AddPoint(100, 1)
    property.SetScalarOpacity(opacityTF)
    renderWindow.Render()

# Connecting the slider widget to the opacity transfer function
slider_widget.AddObserver("InteractionEvent", opacity_control_via_slider)

# Initializing and starting the render window interface
renderWindowInterface.Initialize()
renderWindow.Render()
renderWindowInterface.Start()
```