

Visualization - Lab 2

Oscar Fredriksson

10th June 2021

1 Visualization using Clip

The first part of the lab was to create a visualization using an IsoVolume and Clip filter. I could not find a vtk version of the IsoVolume filter I used in Paraview, so instead I used the contour filter to achieve a similar looking visualization. I started with the same values I used for lab 1, but I had to tweak the values for the clip filter slightly to achieve a similar visualization to what I got in lab 1. I found the filters along with syntax for how to use them by looking through the developer of vtk's list of filters with examples [1]. An image of the visualization can be seen in figure 1 and the written python code can be seen in appendix A.

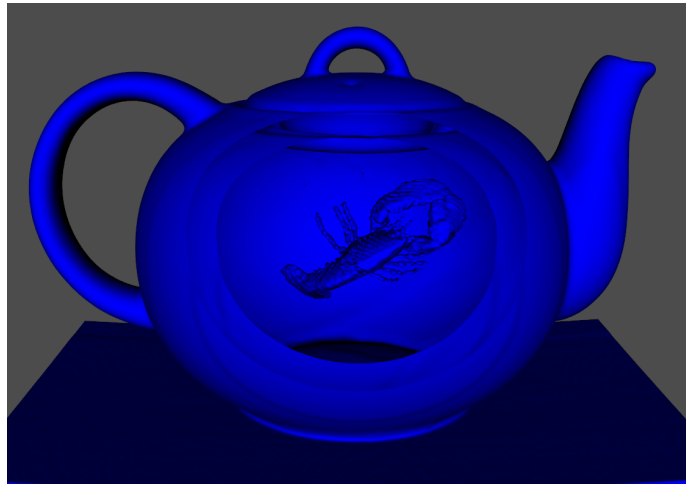


Figure 1: A visualization of the teapot and lobster with a clip to see through the volume.

2 Visualization using Opacity

The second visualization uses two different contour values with different colours to highlight the lobster in yellow. I started with the same values I had in lab 1 for the two contours and the opacity, but had to lower the yellow contour slightly to make the visualization look similar to what was created in lab 1. The created visualization can be seen in figure 2 and the written python code can be seen in appendix B.

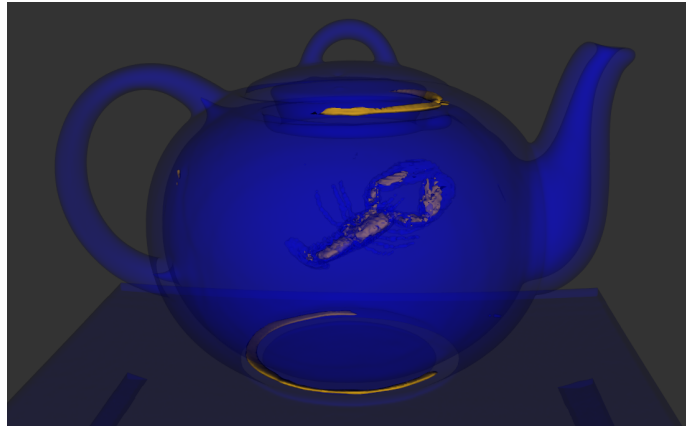


Figure 2: The teapot visualized using contour with values 30 (blue) and 130(yellow) along with an opacity value of 0.2 to see the lobster inside.

3 Conclusions

Two visualizations was created of the teapot that was near identical to what was created using Paraview in lab 1. Figuring out how to use the different vtk filters was a bit challenging, especially since there wasn't much information to find online apart from the examples I found in [1], so I had to rely on these to try and figure out how to use the filters.

After figuring out the syntax vtk worked very similarly to Paraview, and using the same parameters for the filters gave similar results, that after some tweaking could be considered near identical. The conclusions drawn from this lab is therefore that Paraview and vtk are two different tools that can be used to create similar visualizations.

Appendix A Clip visualization code

```
1 import vtk
2
3 # Read raw dataset
4 reader = vtk.vtkImageReader()
5 reader.SetFileName("../dataset/BostonTeapot.raw")
6 reader.SetDataByteOrderToBigEndian()
7 reader.SetNumberOfScalarComponents(1)
8 reader.SetFileDimensionality(3)
9 reader.SetDataExtent(0, 255, 0, 255, 0, 177)
10 reader.SetDataScalarTypeToUnsignedChar()
11 reader.Update()
12
13 contour = vtk.vtkContourFilter()
14 contour.SetInputConnection(reader.GetOutputPort())
15 contour.GenerateValues(2, 30, 255)
16
17
18 plane = vtk.vtkPlane()
19 plane.SetOrigin(160, 150, 140)
20 plane.SetNormal(0.0, -0.35, -0.9)
21
22 clipper = vtk.vtkClipPolyData()
23 clipper.SetInputConnection(contour.GetOutputPort())
24 clipper.SetClipFunction(plane)
25 clipper.SetValue(0)
26 clipper.Update()
27
28 mapper = vtk.vtkDataSetMapper()
29 mapper.SetInputConnection(clipper.GetOutputPort())
30
31 actor = vtk.vtkActor()
32 actor.SetMapper(mapper)
33 actor.GetProperty().SetColor(1.0, 1.0, 1.0)
34
35 renderer = vtk.vtkRenderer()
36 renderer.SetBackground(0.3, 0.3, 0.3)
37 renderer.AddActor(actor)
38
39 render_window = vtk.vtkRenderWindow()
40 render_window.SetWindowName("teapot")
41 render_window.SetSize(800, 800)
42 render_window.AddRenderer(renderer)
43
44 interactor = vtk.vtkRenderWindowInteractor()
45 interactor.SetRenderWindow(render_window)
46 interactor.Initialize()
47 interactor.SetInteractorStyle(vtk.vtkInteractorStyleTrackballCamera())
48 render_window.Render()
49 interactor.Start()
```

Appendix B Opacity visualization code

```
1 import vtk
2
3 # Read raw dataset
4 reader = vtk.vtkImageReader()
5 reader.SetFileName("../dataset/BostonTeapot.raw")
6 reader.SetDataByteOrderToBigEndian()
7 reader.SetNumberOfScalarComponents(1)
8 reader.SetFileDimensionality(3)
9 reader.SetDataExtent(0, 255, 0, 255, 0, 177)
10 reader.SetDataScalarTypeToUnsignedChar()
11 reader.Update()
12
13 contour1 = vtk.vtkContourFilter()
14 contour1.SetInputConnection(reader.GetOutputPort())
15 contour1.GenerateValues(2, 0.0, 30)
16
17 mapper1 = vtk.vtkPolyDataMapper()
18 mapper1.SetInputConnection(contour1.GetOutputPort())
19
20 actor1 = vtk.vtkActor()
21 actor1.SetMapper(mapper1)
22 actor1.GetProperty().SetColor(1.0, 1.0, 1.0)
23 actor1.GetProperty().SetOpacity(0.2)
24
25 contour2 = vtk.vtkContourFilter()
26 contour2.SetInputConnection(reader.GetOutputPort())
27 contour2.GenerateValues(2, 0.0, 130)
28
29 mapper2 = vtk.vtkPolyDataMapper()
30 mapper2.SetInputConnection(contour2.GetOutputPort())
31
32 actor2 = vtk.vtkActor()
33 actor2.SetMapper(mapper2)
34 actor2.GetProperty().SetColor(1, 0.75, 0)
35 actor2.GetMapper().ScalarVisibilityOff()
36
37 renderer = vtk.vtkRenderer()
38 renderer.SetBackground(0.3, 0.3, 0.3)
39 renderer.AddActor(actor1)
40 renderer.AddActor(actor2)
41
42 render_window = vtk.vtkRenderWindow()
43 render_window.SetWindowName("teapot")
44 render_window.SetSize(800, 800)
45 render_window.AddRenderer(renderer)
46
47 interactor = vtk.vtkRenderWindowInteractor()
48 interactor.SetRenderWindow(render_window)
49 interactor.Initialize()
50 interactor.SetInteractorStyle(vtk.vtkInteractorStyleTrackballCamera())
51 render_window.Render()
52 interactor.Start()
```