

Bildverarbeitung Praktikum

Übung 9, Abgabe 02.07.2020

Dr. Christina Gillmann

June 25, 2020

1 Aufgabe 0: Installieren von VTK und laden eines Datensatzes

Für diese Aufgabe benötigen sie eine neue Bibliothek namens VTK (Visualization Toolkit). Mit dieser Bibliothek erhalten sie alle wichtigen Werkzeuge für fortgeschrittene Visualisierungen. Generell bietet VTK sehr viele Algorithmen zur Visualisierung (nicht nur im Rahmen der Bildverarbeitung).

Zur Installation benötigen sie folgenden Befehl:

```
conda install -c anaconda vtk
```

VTK ist ziemlich groß und die Installation kann einige Minuten dauern.

Sie sollen diese Woche mit einem speziellen Datensatz arbeiten. Es handelt sich um den 3D CT scan eines Motors. Ich habe diesen Datensatz (engine.raw) in der Dropbox hinterlegt. Hierbei handelt es sich um ein sehr primitives Datenformat indem alle Werte einfach nur in einer Datei gespeichert sind (ohne header, der zusätzliche Informationen bietet). Deshalb müssen Parameter wie Größe des Bildes und Pixel angegeben werden. Sie finden folgend den Code der ihnen das Laden des Datensatzes ermöglicht.

Den Datensatz können sie wie folgt laden:

```
import vtk

# Read 3D RAW image
reader=vtk.vtkImageReader()
reader.SetDataScalarType(vtk.VTK_UNSIGNED_CHAR) # unsigned int8
reader.SetFileName('engine.raw')
reader.SetNumberOfScalarComponents(1)
reader.SetFileDimensionality(3)
reader.SetDataByteOrderToLittleEndian()
reader.SetDataExtent(0, 255, 0, 255, 0, 127)
reader.SetDataSpacing(1.0, 1.0, 1.0)
reader.Update()
```

2 Aufgabe 1: Volume Rendering

Sie sollen den geladenen Datensatz mit Hilfe des Volume Renderings und einer 1-D Transferfunktion darstellen. Hieryu benötigen sie folgenden Code:

```
renWin = vtk.vtkRenderWindow()
renWin.AddRenderer(ren1)

iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renWin)

# Mit dieser Funktion bestimmen sie die Opacity
# der jeweiligen Grauwerte im Bild
opacityTransferFunction = vtk.vtkPiecewiseFunction()
opacityTransferFunction.AddPoint(20, 0.0)
opacityTransferFunction.AddPoint(255, 0.2)

# Create transfer mapping scalar value to color.
colorTransferFunction = vtk.vtkColorTransferFunction()
colorTransferFunction.AddRGBPoint(0.0, 0.0, 0.0, 0.0)
colorTransferFunction.AddRGBPoint(64.0, 1.0, 0.0, 0.0)
colorTransferFunction.AddRGBPoint(128.0, 0.0, 0.0, 1.0)
colorTransferFunction.AddRGBPoint(192.0, 0.0, 1.0, 0.0)
colorTransferFunction.AddRGBPoint(255.0, 0.0, 0.2, 0.0)

# The property describes how the data will look.
volumeProperty = vtk.vtkVolumeProperty()
volumeProperty.SetColor(colorTransferFunction)
volumeProperty.SetScalarOpacity(opacityTransferFunction)
volumeProperty.ShadeOn()
volumeProperty.SetInterpolationTypeToLinear()

# The mapper / ray cast function know how to render the data.
volumeMapper = vtk.vtkFixedPointVolumeRayCastMapper()
volumeMapper.SetInputConnection(reader.GetOutputPort())

# The volume holds the mapper and the property and
# can be used to position/orient the volume.
volume = vtk.vtkVolume()
volume.SetMapper(volumeMapper)
volume.SetProperty(volumeProperty)

ren1.AddVolume(reader.getOutput())
ren1.ResetCamera()
```

```
renWin.SetSize(600, 600)
renWin.Render()
```

```
iren.Start()
```

a) Machen sie sich mit den Klassen `ColorTransferFunction` und `OpacityTransferFunction` vertraut. Mit diesen sollen sie eine passende Transfer Funktion erstellen. Hierbei ist die Dokumentation von VTK hilfreich

b) Erstellen sie ein Histogramm des Eingangsbildes. Zeigen sie dieses an

c) Erstellen sie eine Transferfunktion mit Hilfe der Parameter in `ColorTransferFunction` und `OpacityTransferFunction`. Sie können dort beliebig viele Punkte für Transparenz und Farben einfügen. Zwischen diesen Punkten wird VTK automatisch interpolieren.

3 Aufgabe 2: Indirect Volume Rendering

In dieser Aufgabe sollen sie den Datensatz durch indirektes Volumen Rendering visualisieren. Hierzu benötigen sie folgenden Code.

```
# Visualization
contour=vtk.vtkMarchingCubes() # vtk.vtkContourFilter()
contour.SetInputConnection(reader.GetOutputPort())
contour.ComputeNormalsOn()
contour.SetValue(0,1)

mapper = vtk.vtkPolyDataMapper()
mapper.SetInputConnection(contour.GetOutputPort())
mapper.ScalarVisibilityOff()

actor = vtk.vtkActor()
actor.SetMapper(mapper)

renderer=vtk.vtkRenderer()
renderer.AddActor(actor)

window = vtk.vtkRenderWindow()
window.SetSize(600, 600)
window.AddRenderer(renderer)

# Create interactor, add window & add observers
interactor = vtk.vtkRenderWindowInteractor()
interactor.SetRenderWindow(window)

# Start renderer & interactor
```

```
window.Render()  
interactor.Initialize()
```

a) Finden sie einen guten Parameter für die Funktion `setValue` im Algorithmus `Marching Cubes`. Hierbei kann ihnen das Histogramm Aufschluss geben.

b) Vergleichen sie die Visualisierungen aus beiden Aufgaben und diskutieren sie welche Vor und Nachteile der jeweilige Ansatz hat.