VTK Introduction

Paulo Dias

deti departamento de electrónica telecomunicações e informática

universidade de aveiro

What is VTK

- December 1993 as supporting softwre software to book “ The visualization Toolkit: An Object Oriented Approach to 3D Graphics by Will Schroeder, Ken Martin and Bill Lorensen (Prentice Hall).

- Maintained by Kitware: http://public.kitware.com/VTK/

- *Software*, books, FAQ’s, exemples, forums…

What is VTK

- open-source software for manipulating and displaying scientific data.
- tools for 3D rendering, a suite of widgets for 3D interaction, and extensive 2D plotting capability
- Core C++ Implementation with multiple wrappers

VTK Main applications

- 3D Computer Graphics
- Visualization
- Image Processing

VTK provides

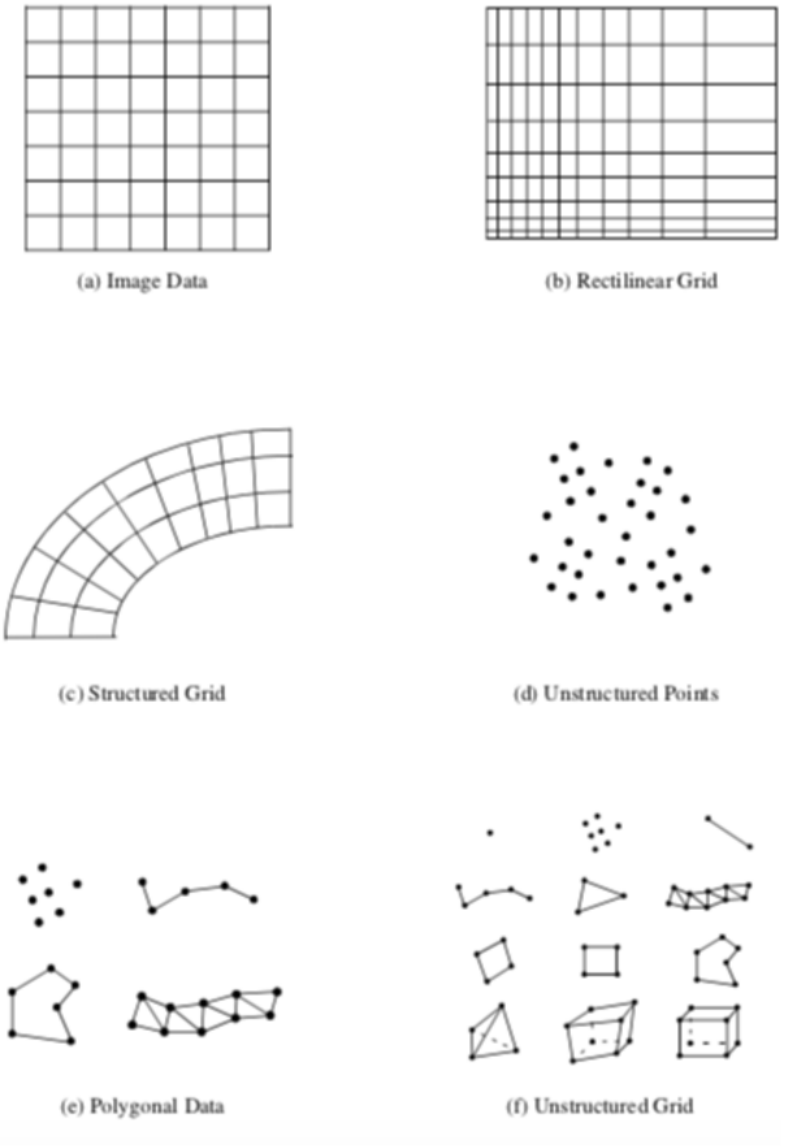
- Data representation: points, meshes, images, volumes, structured and unstructured grids
- Import and exporter for several data format
- Hundreds of filter for data processing

VTK Graphic Model

- VTK use syntax similar to movies: Actors, cameras, lights, properties. The graphic model includes tools for:
  - Window creation and manipulation
  - Mappers and their properties
  - 2D and 3D and volume rendering
  - Lights, cameras and interaction

VTK data Types

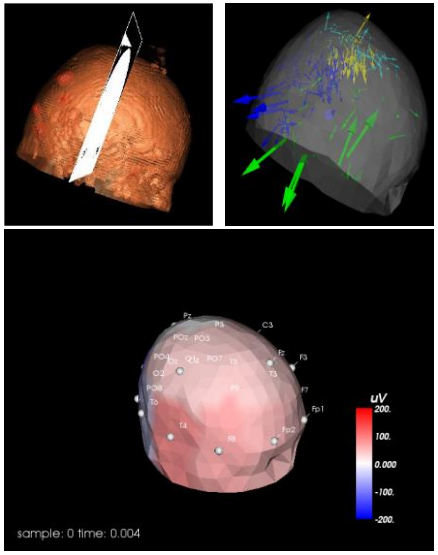
- Polygonal Data
- Image Data
- Rectilinear Grid
- Structured Grid
- Untructured Points
- Unstructured Grid



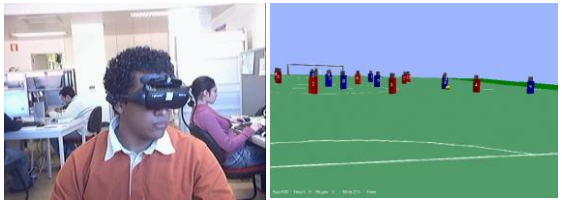
VTK example

```
def main():
    # We Create an instance of vtkConeSource
    coneSource = vtkConeSource()

    # We create an instance of vtkPolyDataMapper to map the polygonal data into graphics primitives.
    coneMapper = vtkPolyDataMapper()
    coneMapper.SetInputConnection( coneSource.GetOutputPort() )

    # We create an actor to represent the cone. The actor orchestrates rendering of the mapper's graphics primitives
    coneActor = vtkActor()
    coneActor.SetMapper(coneMapper)

    # Create the Renderer and assign actors to it
    ren = vtkRenderer()
    ren.AddActor( coneActor )

    # create the render window which will show up on the screen.
    renWin = vtkRenderWindow()
    renWin.AddRenderer(ren)

    renWin.SetWindowName('Cone')

    # loop over 360 degrees and render the cone each time.
    for i in range(0,360):
        renWin.Render()
        # rotate the active camera by one degree
        ren.GetActiveCamera().Azimuth(1)


if __name__ == '__main__':
    main()
```

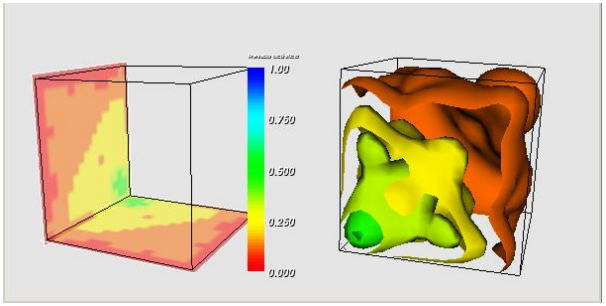
VTK examples UA



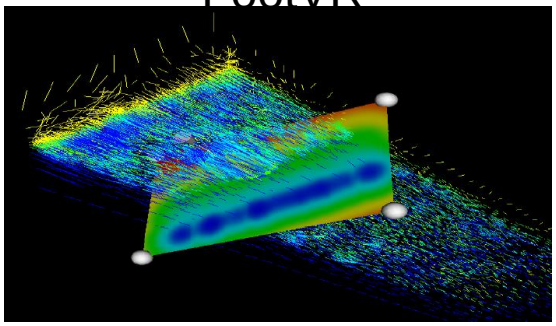
Brain imagery



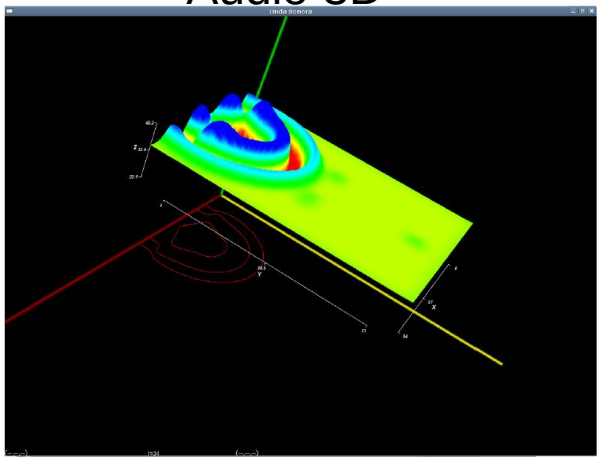
FootVR



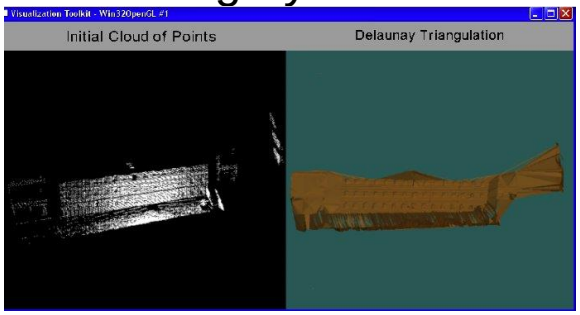
Audio 3D



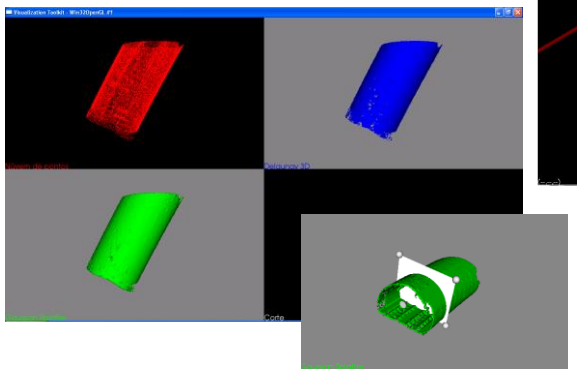
Vectorial data



Audio 2D

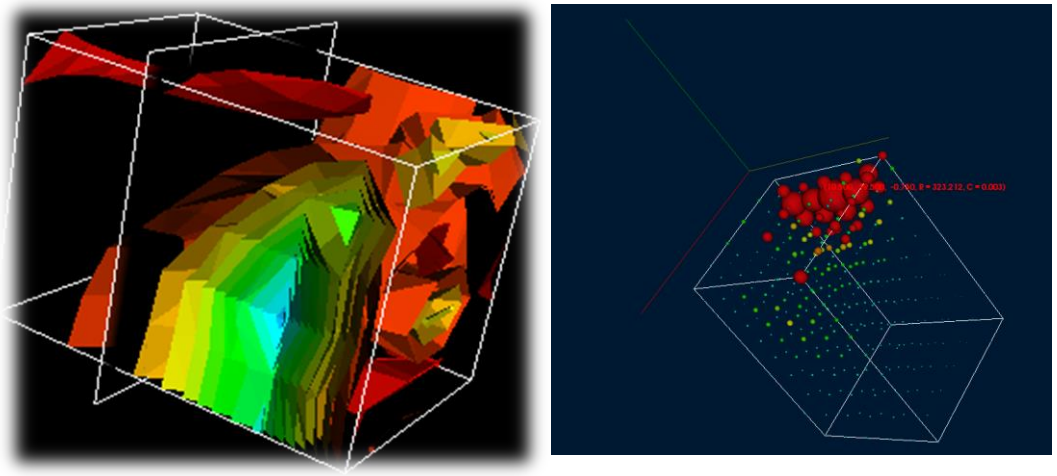


3D meshes

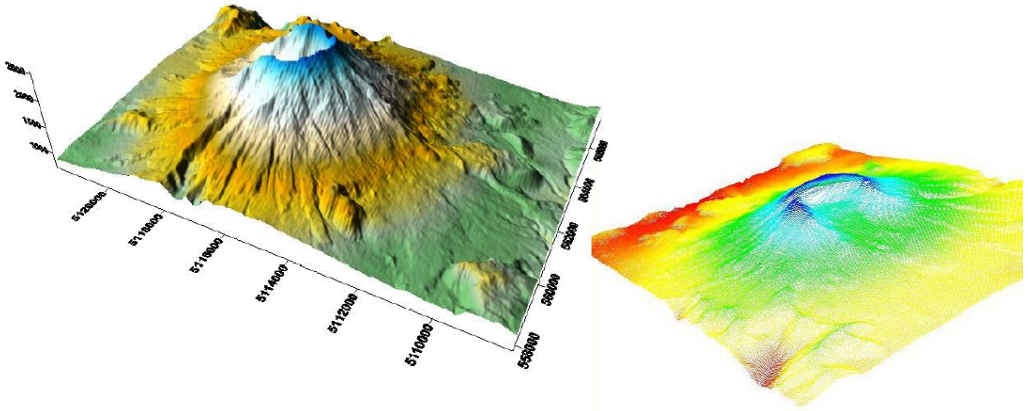


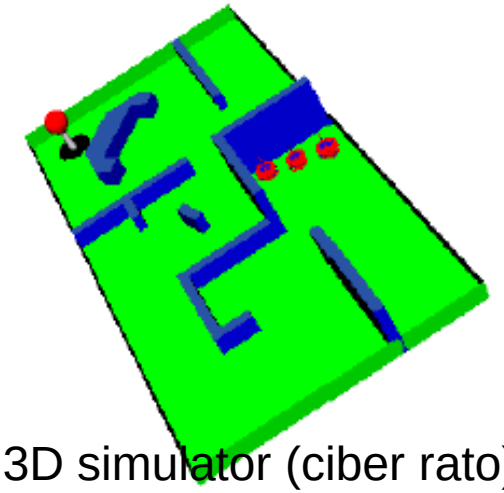
Cuts in models

VTK examples UA

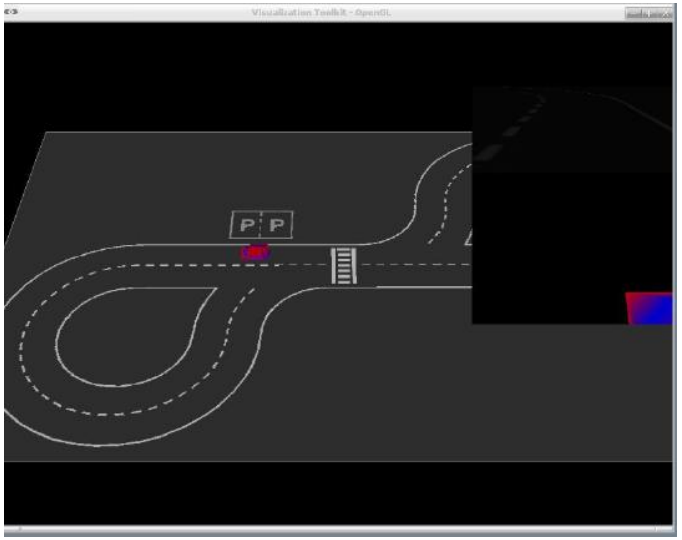


Geocience data



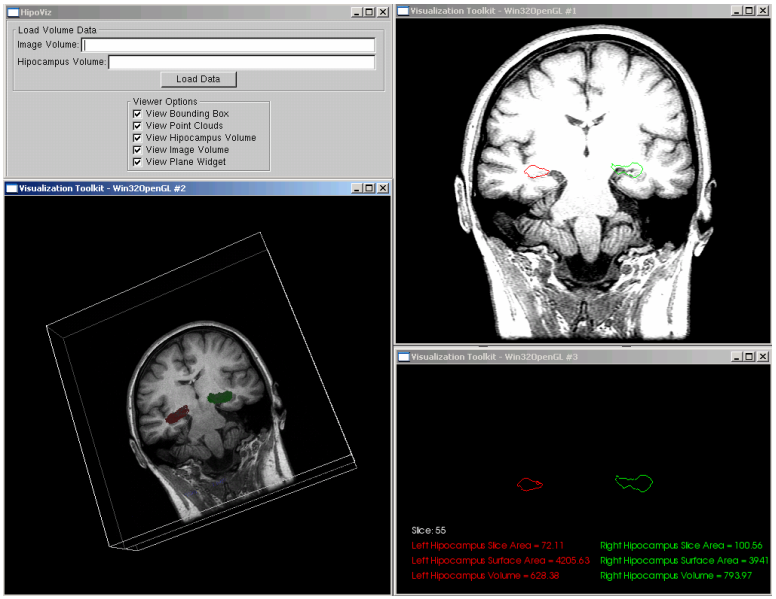


3D simulator (ciber rato)

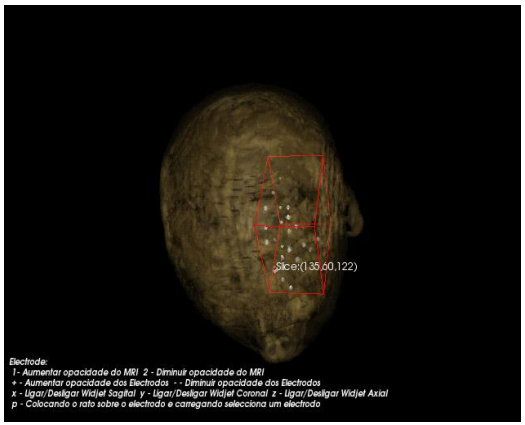


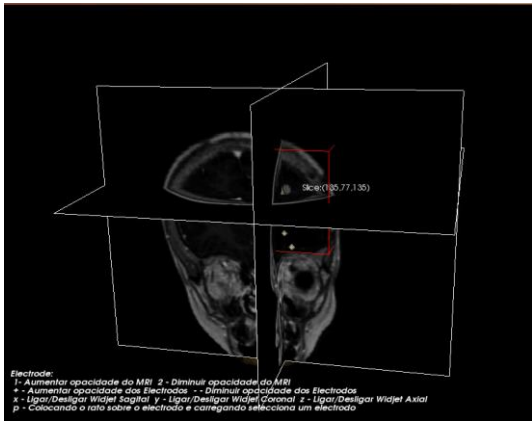
3D simulador (rota)

VTK example

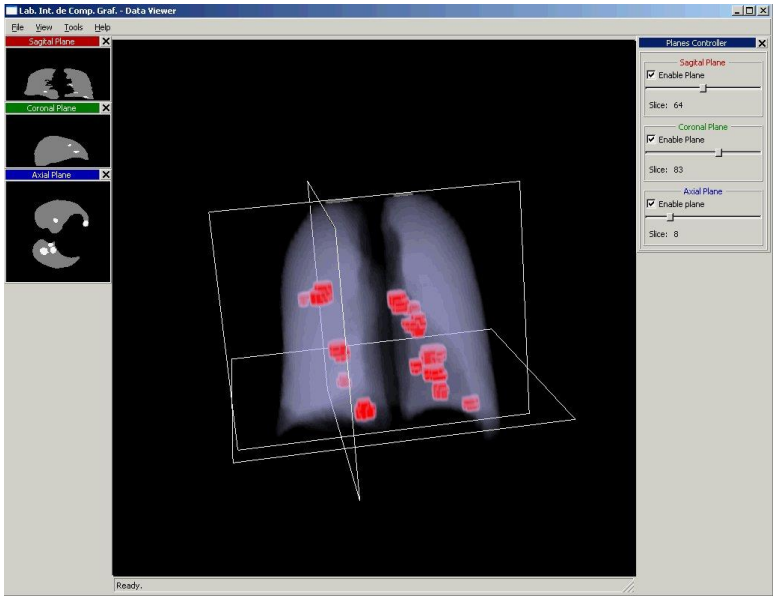


HippoViz: Hipocamp in MR





Electrocorticography



Lung bubble Fox Toolkit e VTK

VTK - Resources

- Schroeder, W., K. Martin, B. Lorensen, *The Visualization Toolkit- An Object Oriented Approach to 3D Graphics*, 2nd ed., Prentice Hall, 1998
- Kitware, inc, *The VTK User's Guide*, Kitware Inc, 2003
- https://vtk.org/