MayaVi: A Free Tool for 3D/2D Data Visualization

Prabhu Ramachandran

October, 25, 2002

Abstract

MayaVi (http://mayavi.sf.net) is an easy to use tool for interactive 3D/2D data visualization and has been developed at the Department of Aerospace Engineering, IIT-Madras. MayaVi is free, Open Source and cross platform. It supports rectilinear, structured and unstructured grids and also polygonal data. MayaVi can visualize grids, scalars, vectors and tensors in different ways. This talk introduces MayaVi, demonstrates its capabilities, features and usage.

1. Overview

Overview

- Introduction
- Features
- History
- Motivational demo
- Installation
- Design
- Creating data files
- Using MayaVi
- Advanced features
- Future directions

2. Introduction

Introduction

- Interactively visualize 2D/3D data.
- Web site: http://mayavi.sf.net
- Hosted at SourceForge.
- Written in Python.
- Uses VTK for the graphics.
- Cross platform (*nix/Linux, Windows and possibly Mac OS X).
- Developed at IITM.
- Users guide available in HTML and PDF.

3. Features

MayaVi Features

- Open Source.
- Graphical user interface.
- Supports rectilinear, structured, unstructured and polygonal data.
- Support for VTK and PLOT3D data.
- Supports scalar, vector and tensor data.
- Save image to PS, BMP, JPEG, PNG, RIB and other files.
- Save the visualization or part of it.
- VRML, 3DS import.

MayaVi Features

- Lookup table editor.
- Data picker, light editor.
- Usable as stand-alone application or as a Python module.
- Scriptable from Python.

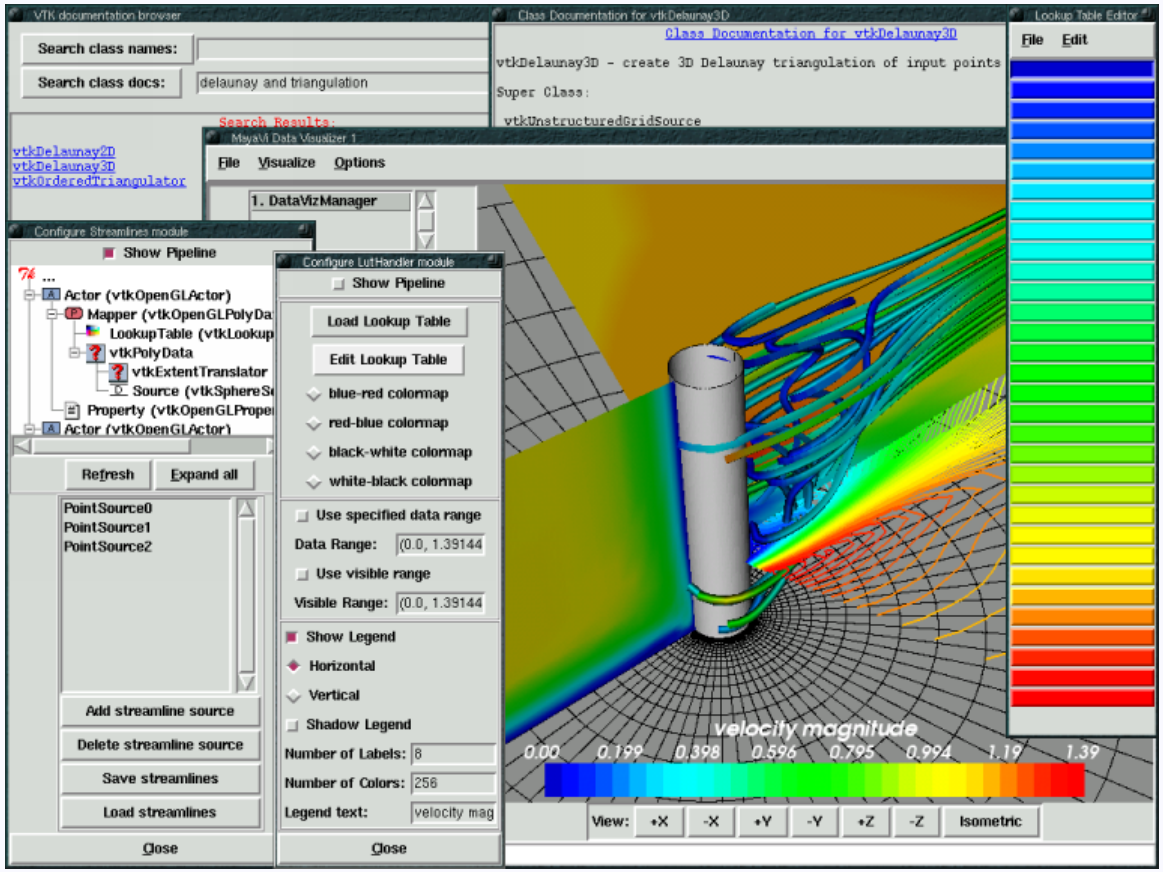
4. History

History

- Started with a few simple scripts in late 1999.
- June 2000 – First release of VTK-CFD.
- January 2001 – Last release of VTK-CFD.
- May 2001 – First release of MayaVi.
- June 2002 – Latest release MayaVi 1.2.

5. Motivational Demo

Demo



Shown here is a visualization for the flow past a cylinder placed on a flat plate. Data courtesy NASA.

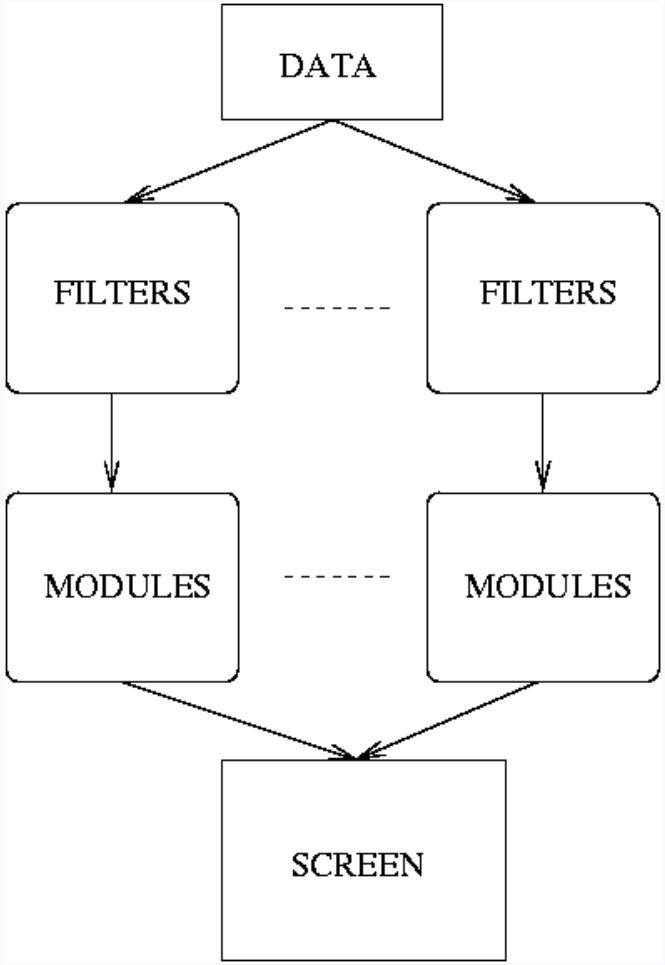
6. Installation

Installation

- Two types of installers.
  1. Binary.
  2. Source.
- Linux – Debian, RPM, tarball.
- Windows – Binary installer, sources.
- Details available at: http://mayavi.sf.net

7. Design

Design



8. Creating data files

VTK data files

- Detailed documentation on this is available here: http://www.vtk.org/pdf/file-formats.pdf.
- VTK data files support the following Datasets.
  1. Structured points.
  2. Rectilinear grid.
  3. Structured grid.
  4. Unstructured grid.
  5. Polygonal data.
- Binary and ASCII files are supported.

General structure

```
# vtk DataFile Version 2.0
A long string describing the file (256 chars)
ASCII | BINARY
DATASET [type]
...

POINT_DATA n
...

CELL_DATA n
...
```

- Point and cell data can be supplied together.
- n is the number of points or cells.

Structured Points

```
# vtk DataFile Version 2.0
Structured points example.
ASCII
DATASET STRUCTURED_POINTS
DIMENSIONS nx ny nz
ORIGIN x0 y0 z0
SPACING sx sy sz
```

- *Important*: There is an implicit ordering of points and cells. The X co-ordinate increases first, Y next and Z last.
- $nx \geq 1, ny \geq 1, nz \geq 1$

Rectilinear Grid

```
# vtk DataFile Version 2.0
Rectilinear grid example.
ASCII
DATASET RECTILINEAR_GRID
DIMENSIONS nx ny nz
X_COORDINATES nx [dataType]
x0 x1 ... x(nx-1)
Y_COORDINATES ny [dataType]
y0 y1 ... y(ny-1)
Z_COORDINATES nz [dataType]
z0 z1 ... z(nz-1)
```

Important: Implicit ordering as in structured points. The X co-ordinate increases first, Y next and Z last.

Structured Grid

```
# vtk DataFile Version 2.0
Structured grid example.
ASCII
DATASET STRUCTURED_GRID
DIMENSIONS nx ny nz
POINTS N [dataType]
x0 y0 z0
x1 y0 z0
x0 y1 z0
x1 y1 z0
x0 y0 z1
...
```

- *Important*: The X co-ordinate increases first, Y next and Z last.
- N = nx*ny*nz

Polygonal data

```
[ HEADER ]

DATASET POLYDATA
POINTS n dataType
x0 y0 z0
x1 y1 z1
...
x(n-1) y(n-1) z(n-1)

POLYGONS numPolygons size
numPoints0 i0 j0 k0 ...
numPoints1 i1 j1 k1 ...
...
```

size = total number of connectivity indices.

Unstructured grids

```
[ HEADER ]

DATASET UNSTRUCTURED_GRID
POINTS n dataType
x0 y0 z0
...
x(n-1) y(n-1) z(n-1)

CELLS n size
numPoints0 i j k l ...
numPoints1 i j k l ...
...

CELL_TYPES n
type0
type1
...
```

size = total number of connectivity indices.

Dataset attributes

- Associated with each point/cell one may specify an attribute.
- VTK data files support scalar, vector and tensor attributes.
- Cell and point data attributes.
- Multiple attributes per same file.

Scalar attributes

```
SCALARS dataName dataType numComp
LOOKUP_TABLE tableName
s0
s1
...
```

- `dataName`: any string with no whitespace (case sensitive!).
- `dataType`: usually `float` or `double`.
- `numComp`: optional and can be left as empty.
- `tableName`: use the value `default`.

Vector attributes

```
VECTORS dataName dataType
v0x v0y v0z
v1x v1y v1z
...
```

- `dataName`: any string with no whitespace (case sensitive!).
- `dataType`: usually `float` or `double`.

Simple example

```
# vtk DataFile Version 2.0
Structured points example.
ASCII
DATASET STRUCTURED_POINTS
DIMENSIONS 2 2 1
ORIGIN 0.0 0.0 0.0
SPACING 1.0 1.0 1.0

POINT_DATA 4
SCALARS Temperature float
LOOKUP_TABLE default
100 200
300 400

VECTORS velocity float
0.0 0.0 0.0
1.0 0.0 0.0
0.0 1.0 0.0
1.0 1.0 0.0
```

9. Using MayaVi

Using MayaVi

- Start MayaVi.
- Open a data file.
- Visualize the data using a Module.
- Optionally filter the data.
- Configure everything to your taste.
- Save the visualization.

10. Advanced features

Advanced features

- Scripting MayaVi from Python.
- Animation.
- Using the Pipeline browser.
- Extending MayaVi.

11. Future directions

Future directions

- A redesign is under way.
- More powerful GUI.
- More modules.
- Should be easier to extend.
- Other GUI toolkit support.