Image processing and computer graphics

Part 3

Adam Sankowski, Marek Kociński

2012

Mayavi 2

Mayavi2 is a general purpose, cross-platform tool for 3-D scientific data visualization. Its features include:

- Visualization of scalar, vector and tensor data in 2 and 3 dimensions
- Easy scriptability using Python
- Easy extendability via custom sources, modules, and data filters
- Reading several file formats: VTK (legacy and XML), PLOT3D, etc.
- Saving rendered visualization in a variety of image formats

Mayavi 2

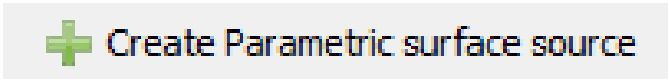
Mayavi2 can be used in three ways:

- completely graphically by using **mayavi2** application
- as a plotting engine from simple Python scripts. The **mlab** scripting API provides a simple way of using Mayavi in batch-processing scripts(**from enthought.mayavi import mlab**)
- Script the Mayavi application from Python:
  - script Mayavi while using the mayavi2 application in order to automate tasks and extend Mayavi's behavior
  - script Mayavi from your own Python based application

Parametric surface

- Parametric surfaces are particularly handy if you are unable to find any data to play with right away. Parametric surfaces are surfaces parametrized typically by 2 variables, u and v
- Let's use parametric surface to show how **mayavi2** application works
- After starting mayavi2, create a simple Parametric surface source by clicking 

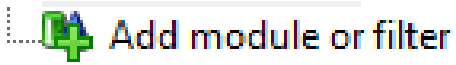
in tree window
- Below you will find different objects you can create. Click on



- Note that you won't see anything visualized on the TVTK scene yet.

Slices 4 to 10 show an examples from Mayavi2 Users Guide written by Prabhu Ramachandran and Gael Varoquaux

Parametric surface

- You can modify the nature of the parametric surface by clicking on the **ParametricSurface** source object.
- To see an outline of the data click on 

  Find Outline in window below and click 

- You should see result in scene window
- To view the actual surface create a Surface module - right-click on your **ParametricSurface**, Add Module, Surface
- You should see a surface. Try buttons above scene window

- You can hide and show modules. For example to hide an Outline right-click on it and find hide/show

Parametric surface

- To view the color legend (used to map scalar values to colors), click on **Colors and Legends** on the tree view and click on the Show legend check-box in window below. This will show you a legend on the TVTK scene. The legend can be moved around on the scene by clicking on it and dragging it. It can also be resized by clicking and dragging on its edges. You can change the nature of the color-mapping by choosing among different lookup tables on the object editor

Parametric surface

- Click on every object in tree window and play with different options below. For example:
  - Change background color in **Mayavi Scene**
  - Change example surface in **ParametricSurface**
  - Change the **Outline** to cornered, or even hide it by right click and Hide/Show
  - Click on **Surface** module in tree window and enable contours. You can also set number of contours and fill them
- You can add as many modules as you like, but remember that not all modules make sense for all data

Reading a file

- We will use heart.vtk file. This is a simple volume of 3D data (32 x 32 x 12 points) with scalars at each point (the points are equally spaced)
- In new Mayavi Scene click on Add Data Source and choose 

from window below
- Example can be found in python directory: …\python\Examples\mayavi-4.1.0\mayavi\data\heart.vtk
- Like previously create an outline
- Add new module by right click on **VTK file**, Add Module, IsoSurface. The iso-surface is colored as per the particular iso-value chosen.

IsoSurface

- Click on **Colors and legends** and Show legend
- Now click on the **IsoSurface** and change iso-value few times. You will see surface of color corresponding to brightness if the image. Compare color with legend.
- After that enable Auto contours. Set Number of Contours to 4 or 5. Now you can see few surfaces corresponding to brightness at the same time
- Create a simple grid plane (right click on **VTK file**, Add Module, Grid Plane) to obtain an idea of the actual points on the grid. Change axis and position a few times

ScalarCutPlane

- To clear the view hide or delete **Grid Plane** by right clicking on it. Then add new module called **ScalarCutPlane**. The ScalarCutPlane module features a very powerful feature called 3D widgets. On the TVTK scene window you will see a cut plane that slices through your data showing you colors representing your data. This cut plane will have a red outline and an arrow sticking out of it. You can click directly on the cut plane and move it by dragging it. Click on the arrow head to rotate the plane. You can also play with scalar warp. When you enable it, you will see a histogram of selected slice.

Mayavi in scripts

- All Mayavi modules, filters and graphics engine are also avaiable in PyLab. You just need to import **mlab** module
- Open PyLab and create new program:

  edit new_program.py
- Now let's try to create six spheres positionned on a line:

  x = [1, 2, 3, 4, 5, 6]

  y = [0, 0, 0, 0, 0, 0]

  z = y

  s = [.5, .6, .7, .8, .9, 1]
- **x**, **y**, and **z** are representing dimensions and **s** is scalar varying

Mayavi in scripts

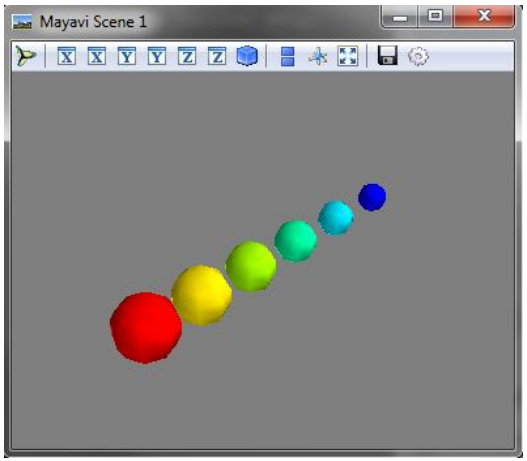
- To visualise spheres we need to import **mlab**

  from enthought.mayavi import mlab

- We represent the dataset as spheres, using points3d(), and the scalar is mapped to diameter of the spheres

  pts = mlab.points3d(x, y, z, s, scale_factor=1)

- Now save program and close it. New window should appear:



Mayavi User Guide

- After that brief preview download full user guide from:
  http://code.enthought.com/projects/mayavi/docs/development/latex/mayavi/mayavi_user_guide.pdf
- You should pay extra attention to *4.2.1* where every visualization module is described
- Read whole chapter *5* to better understand Mayavi scripting
- All datasets are explained in *6.1*
- Test examples from *10.2* and *11.1*