Enjoying the ParaView: Extending & Enhancing ParaView

James Percival

AMCG Seminar Series

19th February 2016

AMCG

- Slides and examples available via

https://github.com/jrper/ParaViewTalk

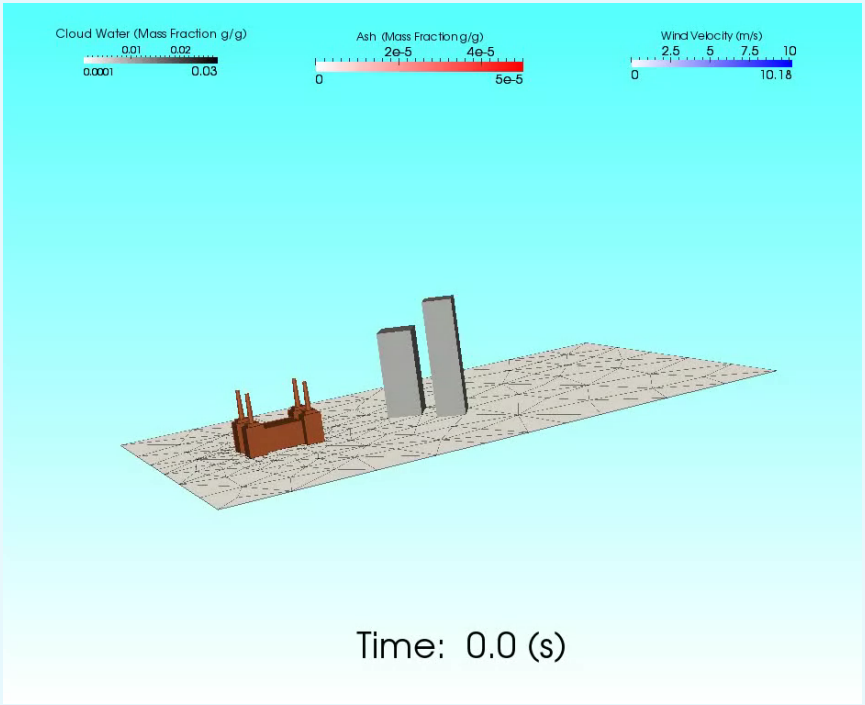
AMCG

Paraview

- “ParaView is an open-source, multi-platform data analysis and visualization application.”
- www.paraview.org
- Developed by Kitware Inc

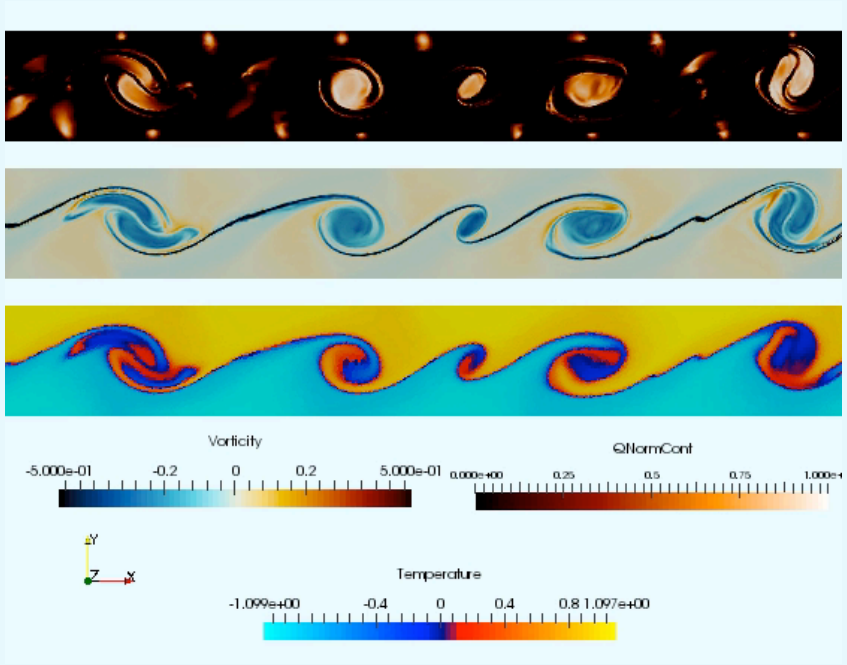
ParaView

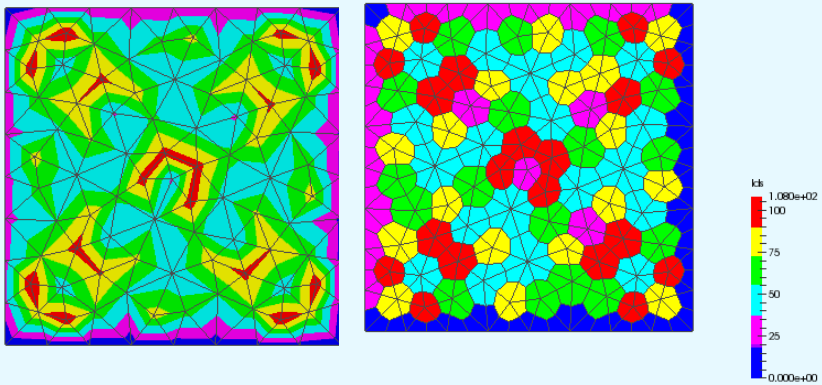
- Used in AMCG/ESE for visualization/post-processing of large unstructured data sets



Why Extend Paraview?

- ParaView isn't perfect
- ParaView is generic tools, applied to specific problems.
- We understand our own data best
- Pretty visualisations help sell good science
- Boring repetition is boring

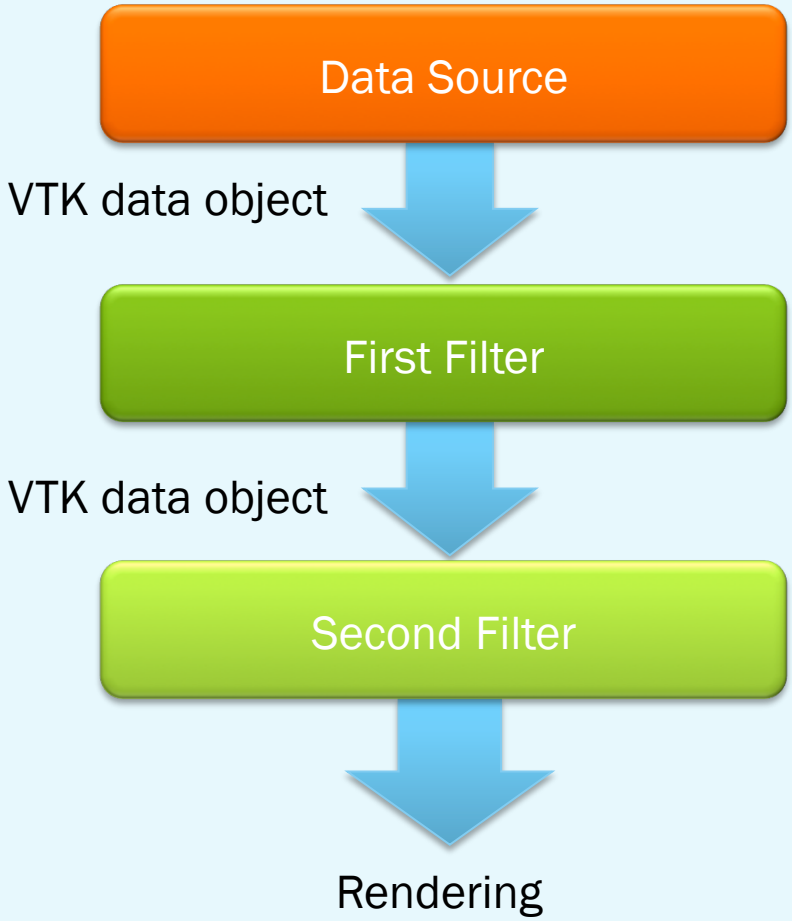




AMCG

The ParaView Data Model

- Pipeline based system
- Sources generate original data
- Chains of filters modify data/view
- Change to one level causes later levels to update themselves.



AMCG

VTK

- Much of ParaView wraps VTK functions, the Visualisation Tool Kit
- www.vtk.org
- “Language”/framework for visualisation
- C++ code with python wrappings

AMCG

VTK

Python wrapping

```
from vtk import *

#Create a source
textSource = vtkTextSource();
textSource.SetText("Hello AMCG!");
textSource.SetForegroundColor(1.0, 0.0, 0.0);
textSource.BackingOn();
textSource.Update();
```

```
from vtk import *

#Create a source
textSource = vtkTextSource();
textSource.SetText("Hello AMCG!");
textSource.SetForegroundColor(1.0, 0.0, 0.0);
textSource.BackingOn();
textSource.Update();

e a mapper and actor
r = vtkPolyDataMapper();
r.SetInputConnection(textSource.GetOutputPort());

 vtkActor();
etMapper(mapper);


e a renderer, render window, and interactor
er = vtkRenderer();
Window = vtkRenderWindow();
Window.AddRenderer(renderer);
WindowInteractor = vtkRenderWindowInteractor();
WindowInteractor.SetRenderWindow(renderWindow);

he actor to the scene
.AddActor(actor);
renderer.SetBackground(1,1,1); # Background color white

#Render and interact
renderWindow.Render();
renderWindowInteractor.Start();
```

AMCG

The ParaView/VTK split

ParaView

- GUI
- Colourmaps
- Labels
- File series
- Graphs
- Screenshots/Animations

VTK

- Data formats
- Drawing stuff
- Geometry
- Maths

Making things prettier

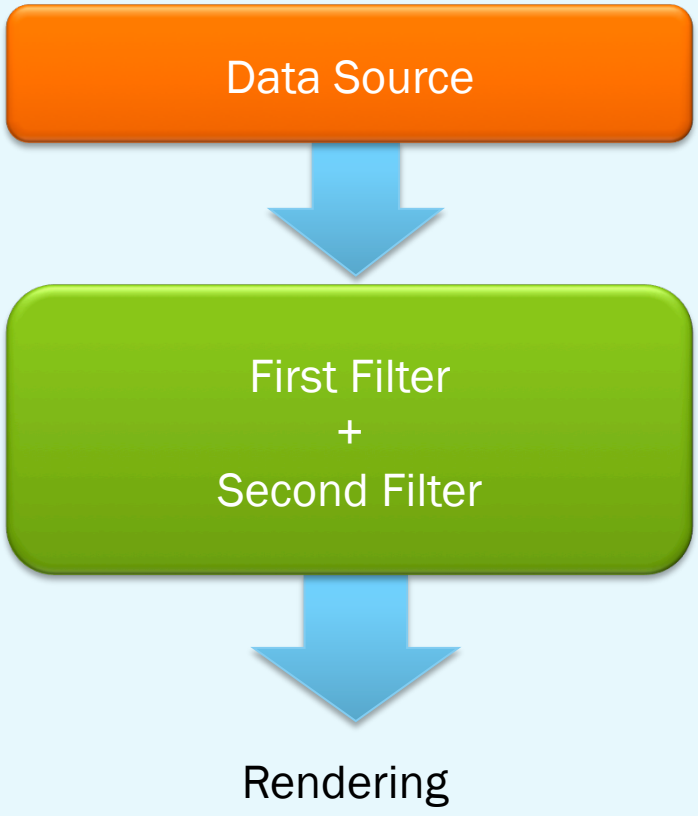
- Use LaTeX in figure text
  - just add $s
- Import your own colour maps

```
<ColorMap space="HSV”  indexedLookup="false” name=”Example">
  <Point x="1" o="1" r="0" g="0" b="1"/>
  <Point x="2" o="1" r="1" g="0" b="1"/>
  <NaN r="0" g="0" b="0"/>
</ColorMap>
```

AMCG

ParaView Custom Filters

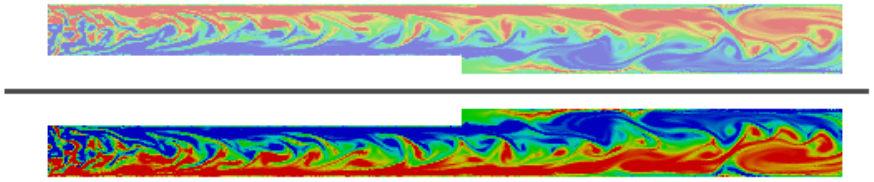
- Quick and easy way to replay the same set of filters onto data.
- Selectable direct from the GUI.
- Can only do stuff you could do eventually anyway.



Example: The Australian Filter

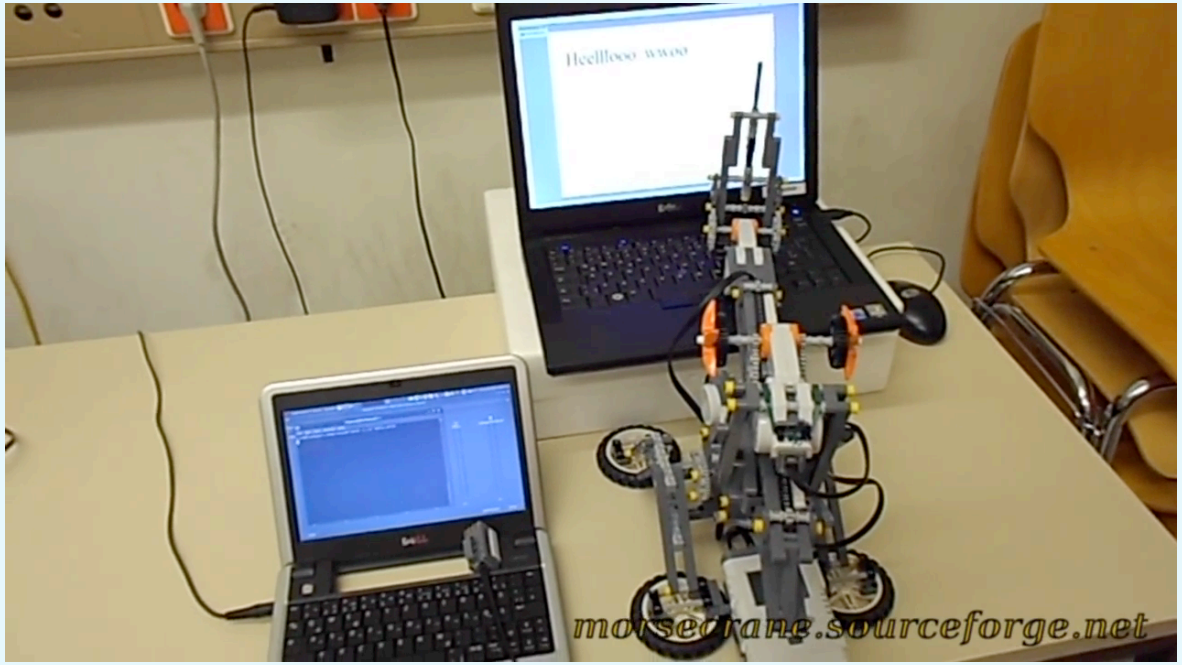
Let's reflect some data to be upside down.

To the live demonstration!



AMCG

ParaView Macros



video: morsecrane.sourceforge.net, CC license

- Playback operations programmatically:
- Need a python script describing the operation
- Script can be recorded from the GUI.
- Can touch almost anything you can

AMCG

ParaView Macros

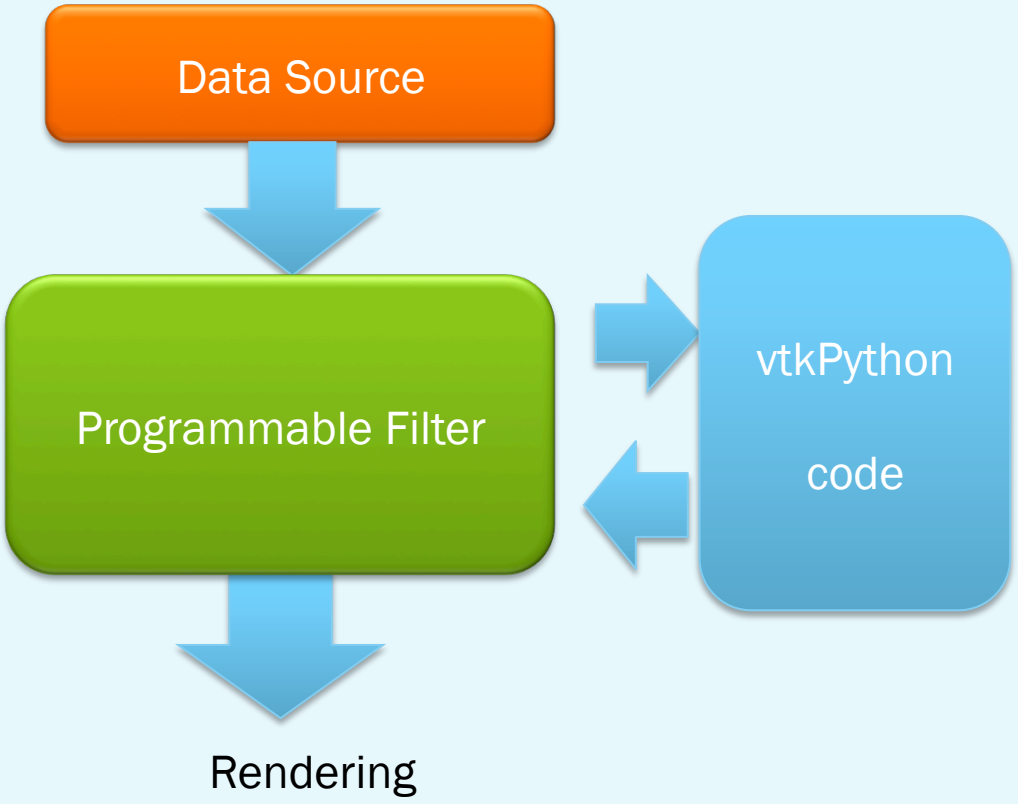
- Example code adds a logo widget from a file to the current view
- Uses a VTK feature not fully wrapped in paraview.

```
from paraview.simple import *
import vtk

image = vtk.vtkPNGReader()
image.SetFileName('AMCGLogo.png')
image.Update()

rep = vtk.vtkLogoRepresentation()
rep.SetImage(image.GetOutput())
intr = GetRenderView().GetInteractor()

widget = vtk.vtkLogoWidget()
widget.SetRepresentation(rep)
widget.SetInteractor(intr)
widget.On()
Render()
```

AMCG

The Python Programmable Filter

- Run arbitrary python code so long, as it produces a VTK object as output
- Modify your data as you see fit
- Australia Mark II



AMCG

The Python Programmable Filter

- “script” run when input data updates
- Australia Mark II
- Written in vtk Python

```
input = self.GetInput()
output = self.GetOutput()

output.DeepCopy(input)

points = output.GetPoints()

x = [0.0,0.0,0.0]
for i in range(points.GetNumberOfPoints()):
    points.GetPoint(i,x)
    x[1] = -x[1]
    points.SetPoint(i,x)

output.GetPointData().DeepCopy(input.GetPointData())
output.GetCellData().DeepCopy(input.GetCellData())
```

AMCG

C++ Plugins

- Highest tech option
- Largest barrier to use.
- Effectively write your own ParaView/VTK source, plus extra XML information to tell the GUI how to work
- Tools>Manage Plugins

```
#include "vtkAustraliaFilter.h"
#include "vtkInformation.h"
#include "vtkInformationVector.h"
#include "vtkPoints.h"
#include "vtkPointData.h"
#include "vtkCellData.h"
#include "vtkUnstructuredGrid.h"
#include "vtkSmartPointer.h"
#include "vtkDataObject.h"
#include "vtkObjectFactory.h"
#include "vtkStreamingDemandDrivenPipeline.h"

vtkCxxRevisionMacro(vtkAustraliaFilter, "$Revision: 0.0$");
vtkStandardNewMacro(vtkAustraliaFilter);

vtkAustraliaFilter::vtkAustraliaFilter(){
  this->SetNumberOfInputPorts(1);
  this->SetNumberOfOutputPorts(1);
};
vtkAustraliaFilter::~vtkAustraliaFilter(){};

int vtkAustraliaFilter::RequestData(
                                                    vtkInformation* vtkNotUsed(request),
                                                    vtkInformationVector **inputVector,
                                                    vtkInformationVector* outputVector )
{

  vtkUnstructuredGrid* input=this->GetUnstructuredGridInput(0);
  vtkUnstructuredGrid* output=this->GetOutput();
  vtkSmartPointer<vtkPoints> points= vtkSmartPointer<vtkPoints>::New();

#ifndef NDEBUG
  this->DebugOn();
#endif

  points->DeepCopy(input->GetPoints());
  double x[3];
  for (vtkIdType j=0; j<points->GetNumberOfPoints(); j++) {
    points->GetPoint(j,x);
    x[1]=-x[1];
    points->SetPoint(j,x);
  }
  output->SetCells(input->GetCellTypesArray(),
                                                    input->GetCellLocationsArray(),
                                                    input->GetCells());
  output->SetPoints(points);
  output->GetCellData()->DeepCopy(input->GetCellData());
  output->GetPointData()->DeepCopy(input->GetPointData());

  return 1;
}
```

AMCG

Anatomy of a plug-in

- ShowCVs
  - CMakeList.txt # CMake instruction file
  - vtkShowCVs.h # C++ header file
  - vtkShowCVs.cxx # C++ source code
  - showCVs.xml # XML for ParaView GUI

AMCG

CMake

- Kitware's build automation tool.
- www.cmake.org
- Like autoconf/ configure

AMCG

CMakeLists.txt

- Example minimal CMakeLists.txt file for a ParaView plugin
- If the code uses other libraries, gets more complicated

```
FIND_PACKAGE(ParaView REQUIRED)
INCLUDE(${PARAVIEW_USE_FILE})

ADD_PARAVIEW_PLUGIN(
  AustraliaFilter "0.0"
  SERVER_MANAGER_XML
AustraliaFilter.xml
  SERVER_MANAGER_SOURCES
vtkAustraliaFilter.h vtkAustraliaFilter.cxx
  )
```

AMCG

AustraliaFilter.xml

- GUI xml file tells ParaView the contract your plugin provides
- Sets the widgets shown in properties tab.
- Also allows documentation.

```
<ServerManagerConfiguration>
  <ProxyGroup name="filters">
    <SourceProxy name="vtkAustraliaFilter"
                 class="vtkAustraliaFilter"
                 label="Australia Filter">
      <Documentation long_help="AustraliaFilter Filter"
                     short_help="Australia Filter">
      </Documentation>
      <InputProperty
          name="Input"
          command="SetInputConnection">
        <ProxyGroupDomain name="groups">
          <Group name="sources"/>
          <Group name="filters"/>
        </ProxyGroupDomain>
        <DataTypeDomain name="input_type">
          <DataType value="vtkUnstructuredGrid"/>
        </DataTypeDomain>
      </InputProperty>
    </SourceProxy>
  </ProxyGroup>
</ServerManagerConfiguration>
```

AMCG

the core of the plugin

```
int vtkAustraliaFilter::RequestData(
                        vtkInformation* vtkNotUsed(request),
                        vtkInformationVector **inputVector,
                        vtkInformationVector* outputVector )
{
 vtkUnstructuredGrid* input=this->GetUnstructuredGridInput(0);
 vtkUnstructuredGrid* output=this->GetOutput();
 vtkSmartPointer<vtkPoints> points=vtkSmartPointer<vtkPoints>::New();
 points->DeepCopy(input->GetPoints());
 double x[3];
 for (vtkIdType j=0; j<points->GetNumberOfPoints(); j++) {
     points->GetPoint(j,x);
     x[1]=-x[1];
     points->SetPoint(j,x);
 }
 output->SetCells(input->GetCellTypesArray(),
                  input->GetCellLocationsArray(),
                  input->GetCells());
 output->SetPoints(points);
 output->GetCellData()->DeepCopy(input->GetCellData());
 output->GetPointData()->DeepCopy(input->GetPointData());

 return 1;
}
```

AMCG

Summary

Extension Method	Ease of use	Ease of creation	Power	Repeatable
Custom Filter	Easiest	Easy	Limited	Easy
Python Macro	Easy	Moderate	Do lots	Limited
Python Programmable Filter	Moderate	Moderate	Do lots	Easy
C++ plugin	Moderate	Hard	Do anything	Easy

QUESTIONS & COMMENTS

AMCG

Where to go next:

- ParaView internal help documentation
- Paraview website & wiki:
  - www.paraview.org/Wiki/ParaView
- vtk documentation:
  - http://www.vtk.org/doc/nightly/html/
- My ParaView Plugins
- Your colleagues