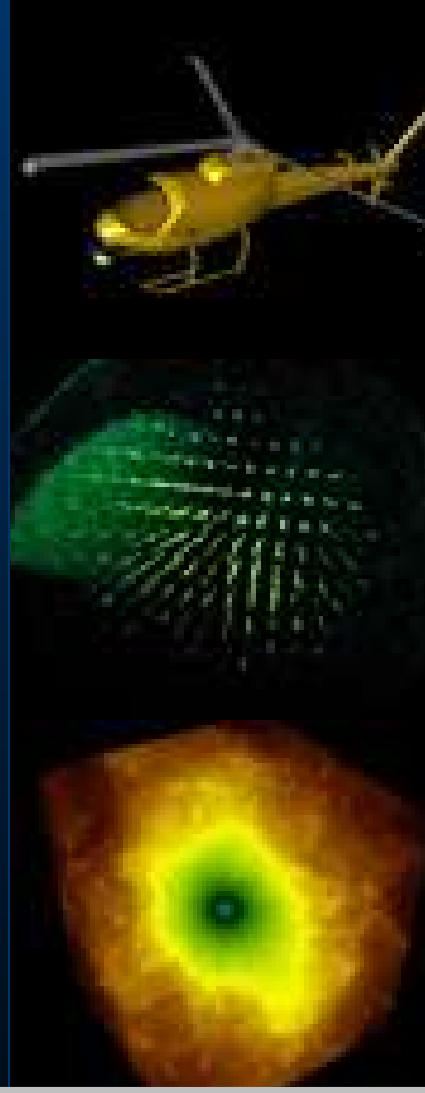
Visualization ToolKit (VTK)
Part I



Weiguang Guan

RHPCS, ABB 131-G
Email: guanw@mcmaster.ca
Phone: 905-525-9140 x 22540

McMaster University

Outline

- Overview
- Installation
- Typical structure of a VTK application
- Visualization pipeline
- VTK scene

Overview

McMaster University

- Higher-level visualization library or API
- Open source – www.vtk.org
- Major contributors
  - Kitware inc
  - GE
  - Sandia National Lab
  - Los Alamos National Lab
- Thousands users from industry and research institutes

Overview

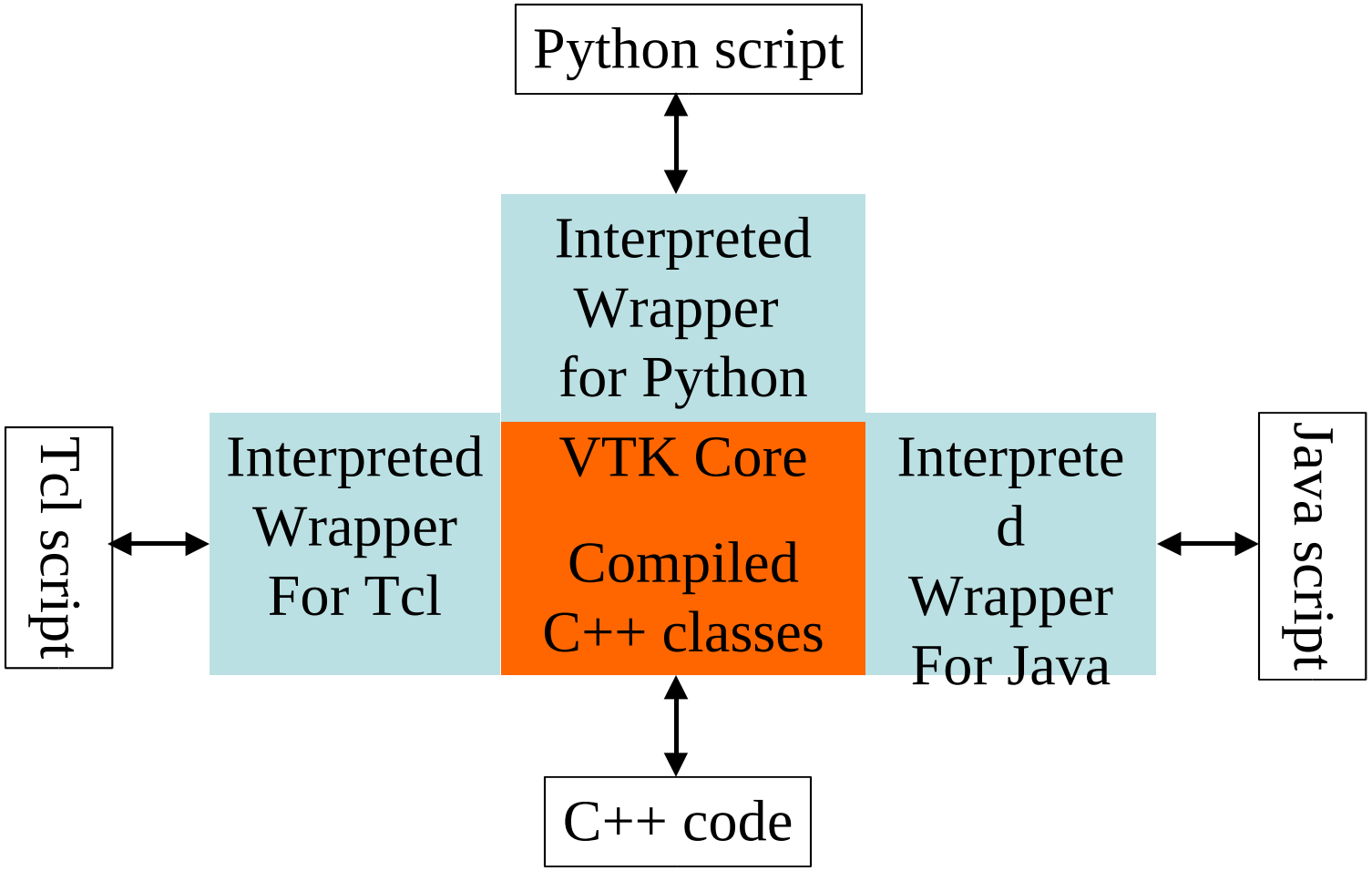
McMaster University

- Multiple platforms – Unix, MacOs, Windows
- 1130 C++ classes
- Java, Python, Tcl wrappers
  - Performance compared with using VTK's C++ interface
  - Faster development cycle
  - Fast prototyping
- Last release 5.0.2

Overview

McMaster University

- **Architecture**



Overview

McMaster University

- Documentation
  - Online manual: www.vtk.org/doc/release/5.0/html/
  - The Visualization Toolkit User's Guide
  - The Visualization Toolkiit, An Object-Oriiented Approach To 3D Graphics
  - Mailing list: vtkusers@vtk.org
  - VTK wiki: www.vtk.org/Wiki/VTK
  - Professional supports from Kitware inc

Overview

McMaster University

- Related software
  - ITK: www.itk.org
  - CMake: www.cmake.org
  - ParaView: www.paraview.org

Installation

McMaster University

- VTK Source directory
  - VTK/Common – Core classes
  - VTK/Filtering – Pipeline implementation superclasses
  - VTK/GenericFiltering – Adaptor framework
  - VTK/Graphics – Filter classes that process 3D data
  - VTK/Hybrid – Complex classes depending on graphics & imaging
  - VTK/Imaging – Image processing filters
  - VTK/IO – Reading/Writing data

Installation

McMaster University

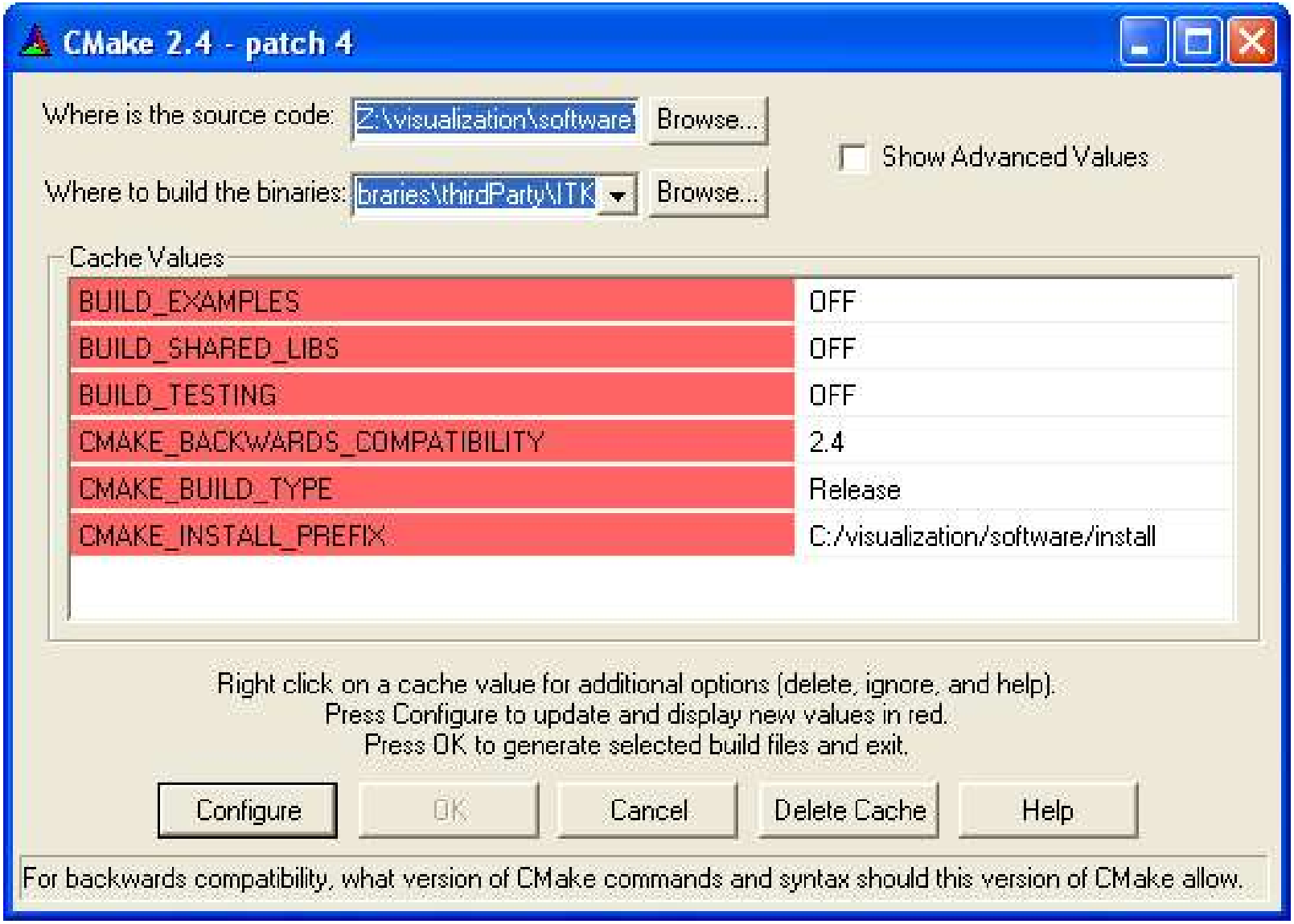
- VTK/Paralell – Paralell processing
- VTK/Rendering – Rendering geometric data
- VTK/Utilities – Utility libraries
- VTK/VolumeRendering – Rendering volume data
- VTK/Widgets – Interactive graphical objects
- VTK/Wrapping – Script language binder

Installation

McMaster University

- CMake: www.cmake.org
  - Cross-platform
  - Open Source
  - Out-of-place build
- CMake configure for VTK
  - Wrapper for Java, Python, Tcl
  - Shared or static lib
  - Specify C/C++ compilers
  - Specify path include paths and libs

Installation

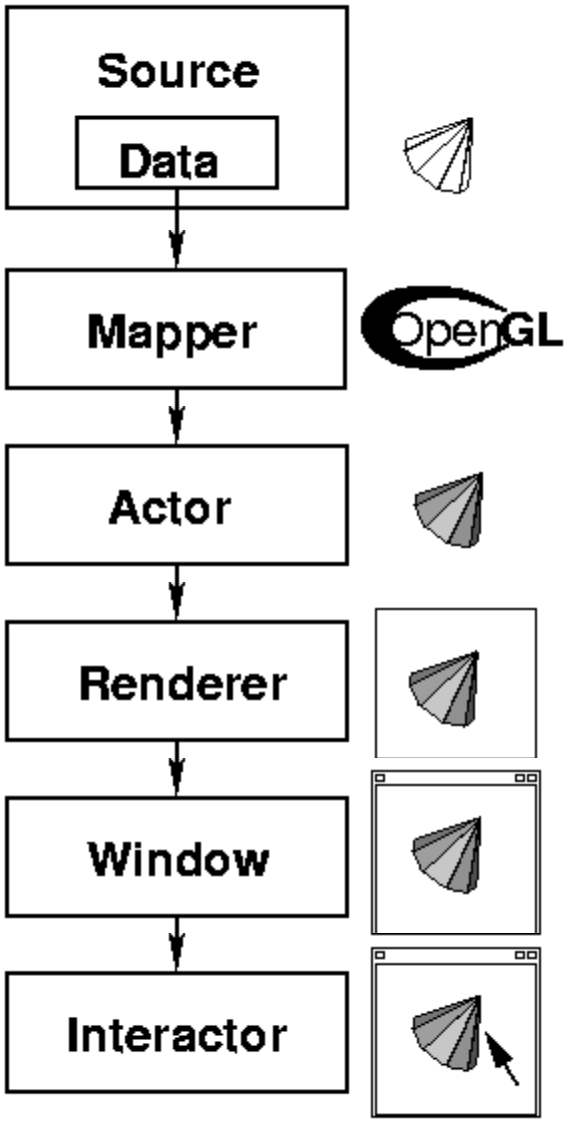


Installation

McMaster University

- Usage:
  - cd build_dir
  - ccmake source_dir
  - Configure VTK build with cmake
  - make
  - make install

Typical Structure



package require vtk

vtkConeSource cone

cone SetHeight 3.0

vtkPolyDataMapper coneMapper

coneMapper SetInputConnection [cone GetOutputPort]

vtkActor coneActor

coneActor SetMapper coneMapper

vtkRenderer ren1

ren1 AddActor coneActor

vtkRenderWindow renWin

renWin AddRenderer ren1

renWin SetSize 300 300

vtkRenderWindowInteractor iren

iren SetRenderWindow renWin

iren Initialize

Typical Structure

McMaster University

Tcl script:

```
package require vtk
vtkConeSource cone
cone SetHeight 3.0

vtkPolyDataMapper coneMapper
coneMapper SetInputConnection
[cone GetOutputPort]

vtkActor coneActor
coneActor SetMapper coneMapper

vtkRenderer ren1
ren1 AddActor coneActor

vtkRenderWindow renWin
renWin AddRenderer ren1
renWin SetSize 300 300

vtkRenderWindowInteractor iren
iren SetRenderWindow renWin
iren Initialize
```

C++ code:

```
#include “vtkConeSource.h”
#include “vtkPolyDataMapper.h”
#include “vtkActor.h”
#include “vtk

vtkConeSource *cone =
      vtkConeSource::New();
Cone->SetHeight(3.0);

vtkPolyDataMapper *coneMapper;
coneMapper = vtkPolyDataMapper::New();
coneMapper->SetInputConnection
(cone->GetOutputPort());

……
……
```

Visualization Pipeline

McMaster University

- Data
  - Dimensions: 2D, 3D, 4D …
  - Type:
    - Geometric data (points, lines, polygons)
    - Discrete data (samplings of properties in n-D space)
  - Data properties:
    - Scalar (density, temperature, etc)
    - Vector (velocity, momentum, etc)
    - Tensor

Visualization Pipeline

- Topology and geometry of sampling data

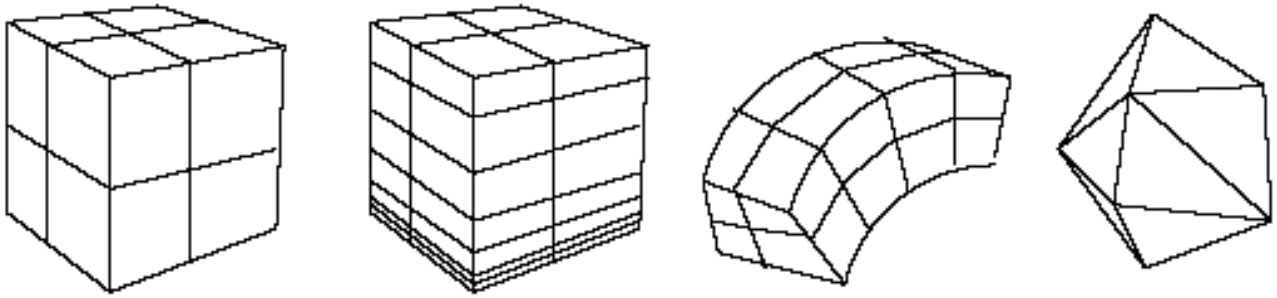
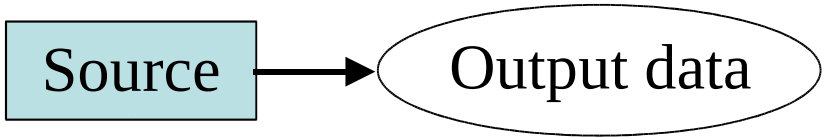


	Image data	Rectilinear	Curvilinear	Unstructured
Topology	regular	regular	regular	irregular
Geometry	regular	partially regular	irregular	irregular

Visualization Pipeline

McMaster University

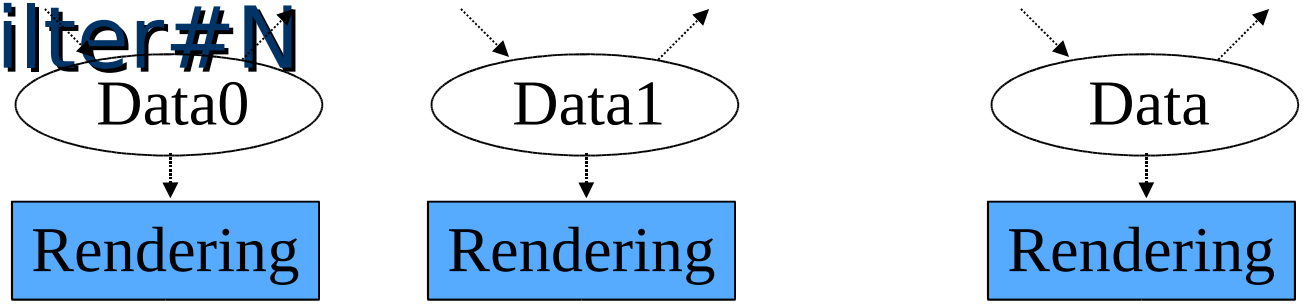
- Dataset is a viewable item
- Processing modules
  - Source – data file reader or graphical primitives



  - Filter – processing module

Visualization Pipeline

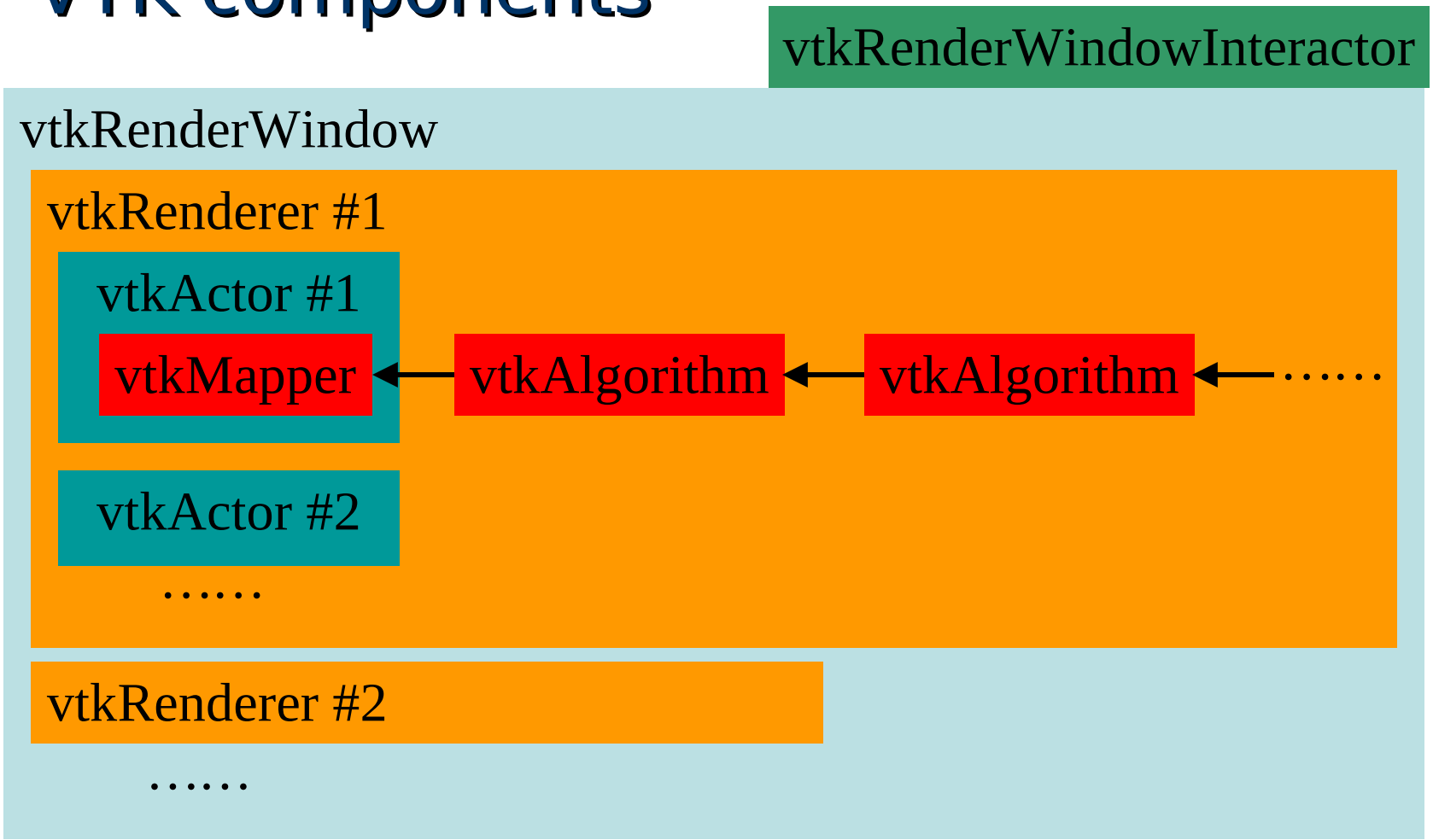
- Visualization pipeline:
  Source Filter#1 Filter#2 ... Filter#N




```
Filter_1->SetInputConnection(Source->GetOutputPort());
Filter_2->SetInputConnection(Filter_1->GetOutputPort());
.....
```

VTK Scene

- **VTK components**



VTK Scene

McMaster University

- vtkRenderWindowInteractor
  - **Set render window**
  - Set interactor style
  - Set desired still/interactive refresh rates
  - Control cursor

VTK Scene

McMaster University

- vtkRenderWindow
  - **Add/Remove renderers**
  - Set window size/position
  - Control stereo setting
  - Enable/Disable anti-aliasing
  - Control cursor
  - Set desired still/interactive refresh rates

VTK Scene

McMaster University

- vtkRenderer
  - **Add/Remove actors**
  - Add/Remove lights
  - Set active camera
  - Enabel/Disable storing rendered image for update

VTK Scene

McMaster University

- vtkProp (superclass of vtkActor2D and vtkProp3D)
  - **Set mapper**
  - Set visibility
  - Set orientation/location/scaling
  - Set display style
  - Set surface property

VTK Scene

McMaster University

- vtkAbstractMapper3D
  - Subclasses
    - vtkPolyDataMapper: for polygonal data
    - vtkVolumeMapper: for rectilinear volumetric data
    - vtkUnstructuredGridVolumeMapper: for unstructured data
  - Map input data into graphics primitives

VTK Scene

McMaster University

- vtkAlgorithm
  - Number of inputs/outputs
    - Source: 0 input, >1 outputs
      - Procedural objects (cube, cone, cylinder…)
      - From data files
    - Filter: >1 inputs, >1 outputs
      - Polygonal Polygonal
      - Polygonal Volumetric
      - Volumetric Polygonal
      - Volumetric Volumetric
    - Writer: >1 inputs, 0 output
      - To data files

VTK Scene

McMaster University

- Examples
  - Tutorial/Step5/Tcl/Cone5.tcl
  - VolumeRendering/Tcl/SimpleRaycast.tcl