ITK and VTK-
The Standard Libraries

Eduard Schreibmann

Department of Radiation Oncology and Winship Cancer Institute, Atlanta, GA

Who Should Use ITK ?

- Best suited for :
  - Larger research projects
  - Need customization/changes over time
  - Likely to implement new algorithm or approaches
- Not suited for:
  - Projects where compiled software exists already.

ITK

Kitware

Search

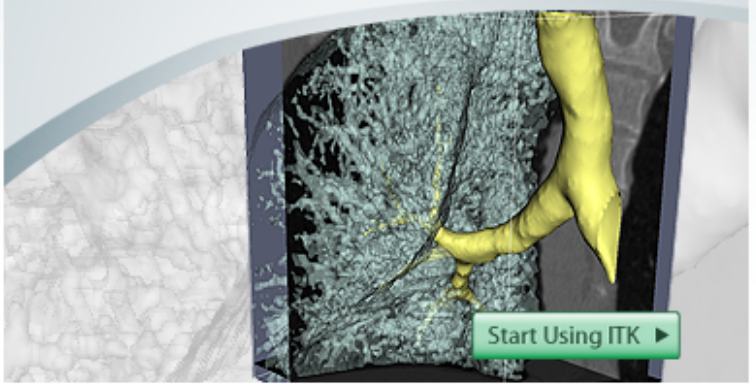
PROJECT RESOURCES HELP OPEN SOURCE

Welcome to the National Library of Medicine **Insight Segmentation and Registration Toolkit (ITK).** ITK is an open-source, cross-platform system that provides developers with an extensive suite of software tools for image analysis. Developed through extreme programming methodologies, ITK employs leading-edge algorithms for registering and segmenting multidimensional data. The goals for ITK include:

- Supporting the Visible Human Project.
- Establishing a foundation for future research.
- Creating a repository of fundamental algorithms.
- Developing a platform for advanced product development.
- Support commercial application of the technology.
- Create conventions for future work.
- Grow a self-sustaining community of software users and developers.

ITK

ITK provides leading-edge segmentation and registration algorithms in two, three, and more dimensions; it is distributed as an open-source software package.



NSF NLM NIH NIDCD NIDCR

My Toolbox

Rapid prototyping

Repository of image processing algorithms

Visualize images

Repository of visualization algorithms

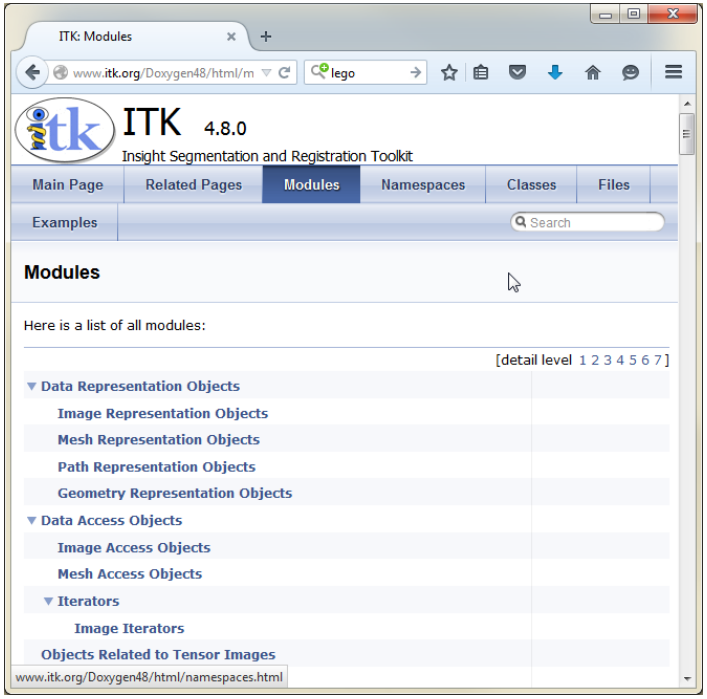
Visualize meshes

Multi-platform build system

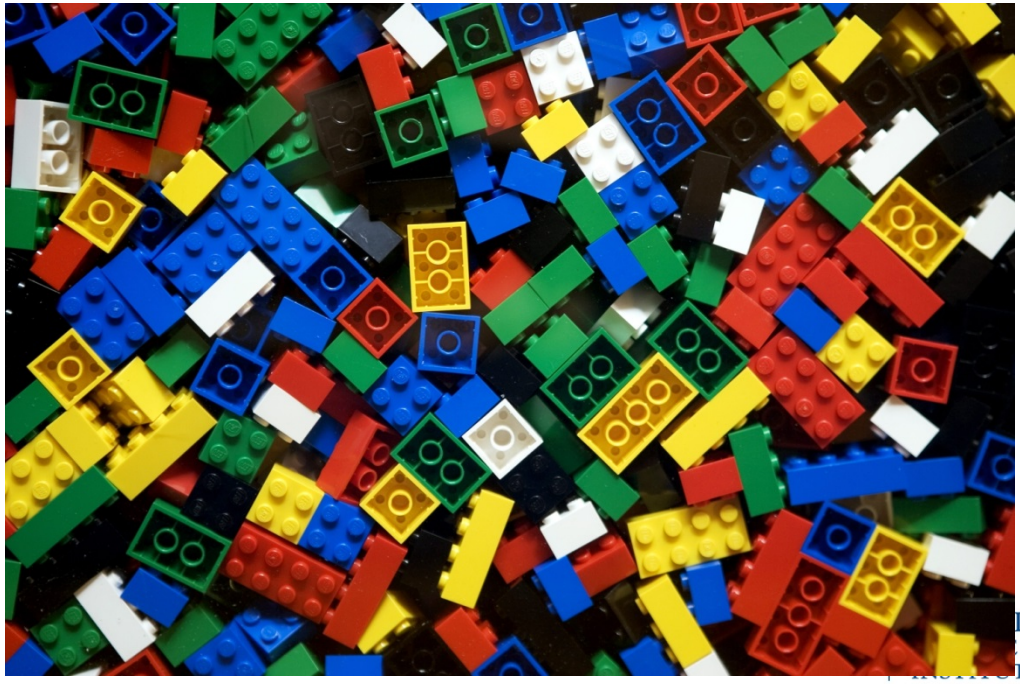
Why I Like It ?

- ITK and VTK are like Legos
  - ITK designed as building blocks from which customized applications are created.

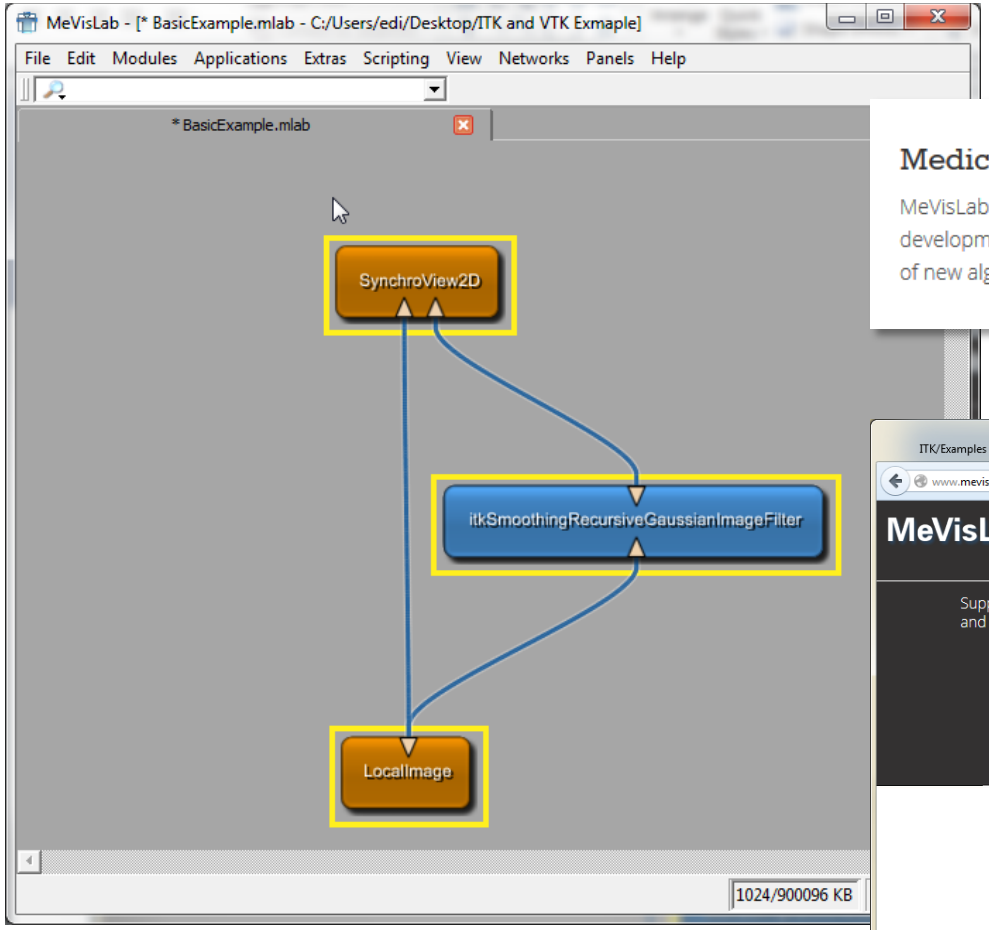
My Playground



Son's Playground

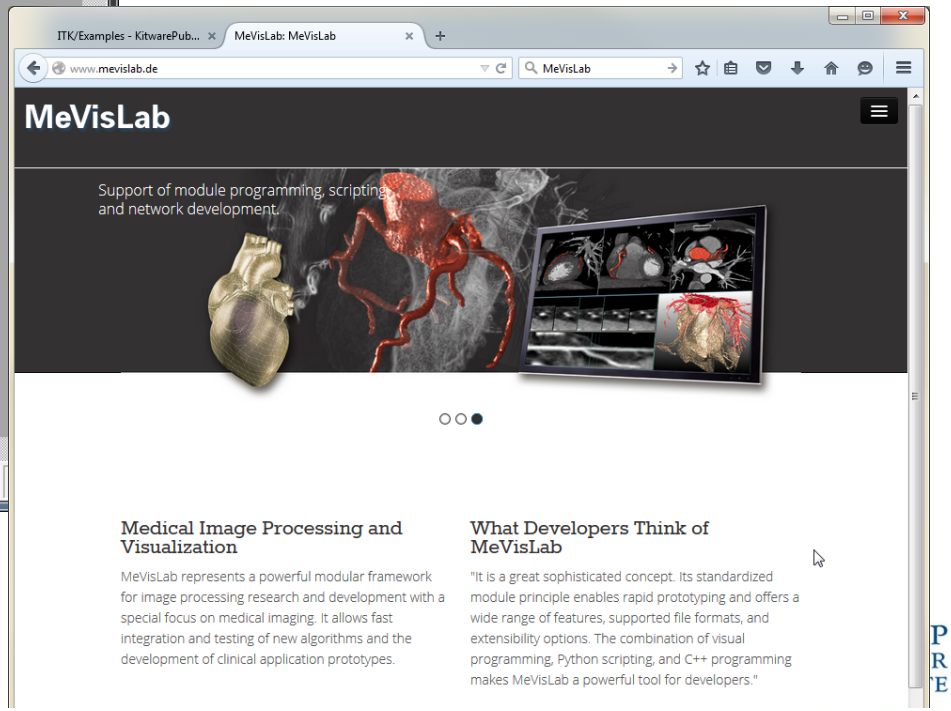


Prototyping



Medical Image Processing and Visualization

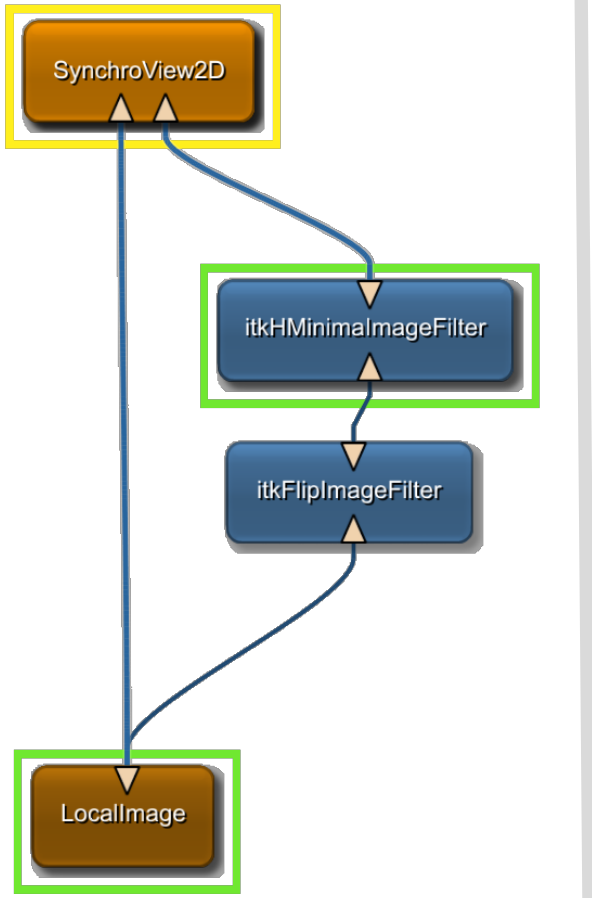
MeVisLab represents a powerful modular framework for image processing research and development with a special focus on medical imaging. It allows fast integration and testing of new algorithms and the development of clinical application prototypes.

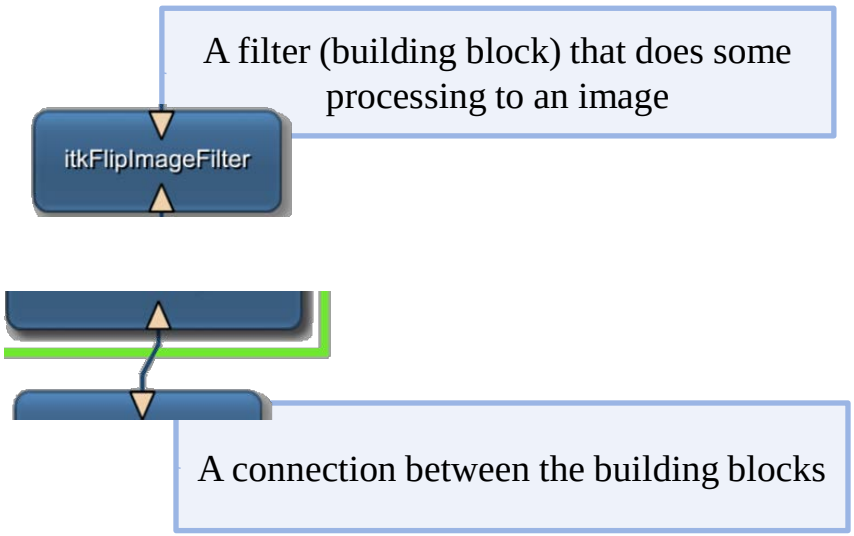


Drag-drop interface to create customized algorithms from building blocks

Why Lego Analogy ?

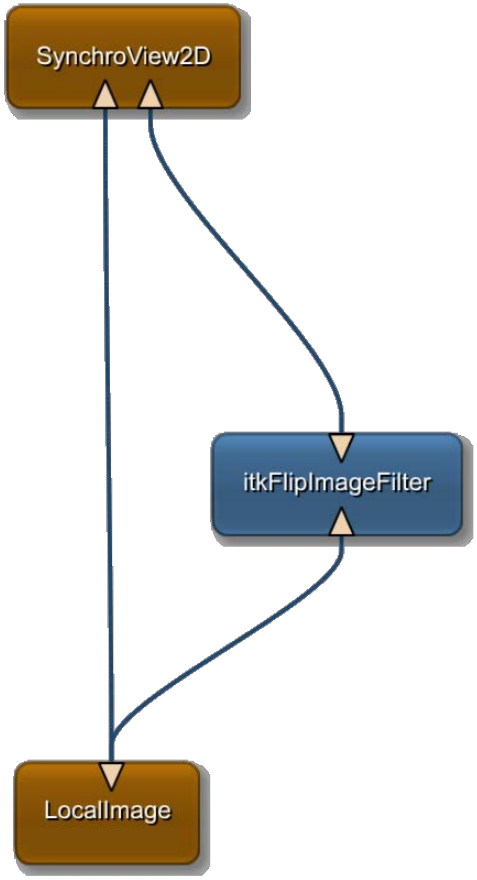
Libraries are a collection of “building blocks” that can be connected to **your** needs.

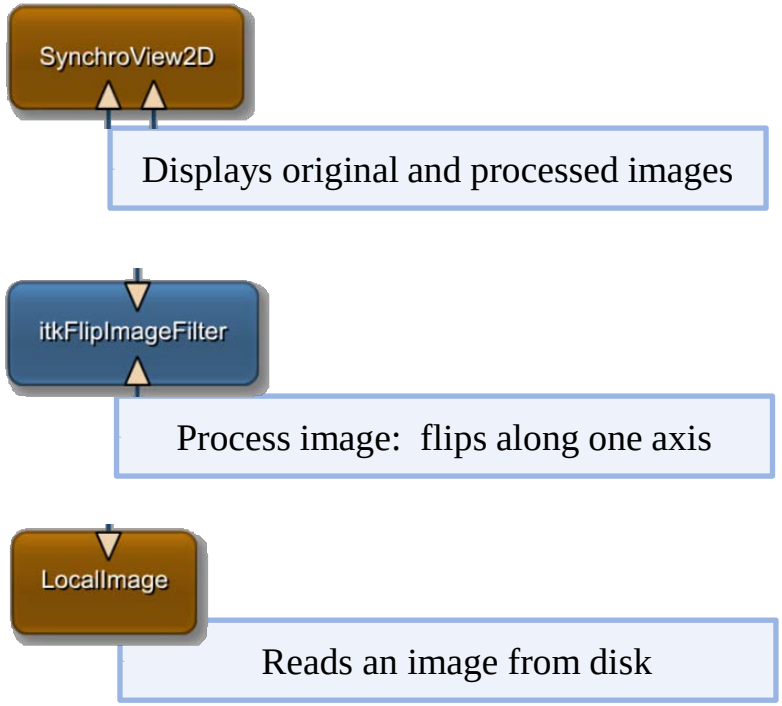




Example

The itkFlipImageFilter will flip an image along a user specified axis.

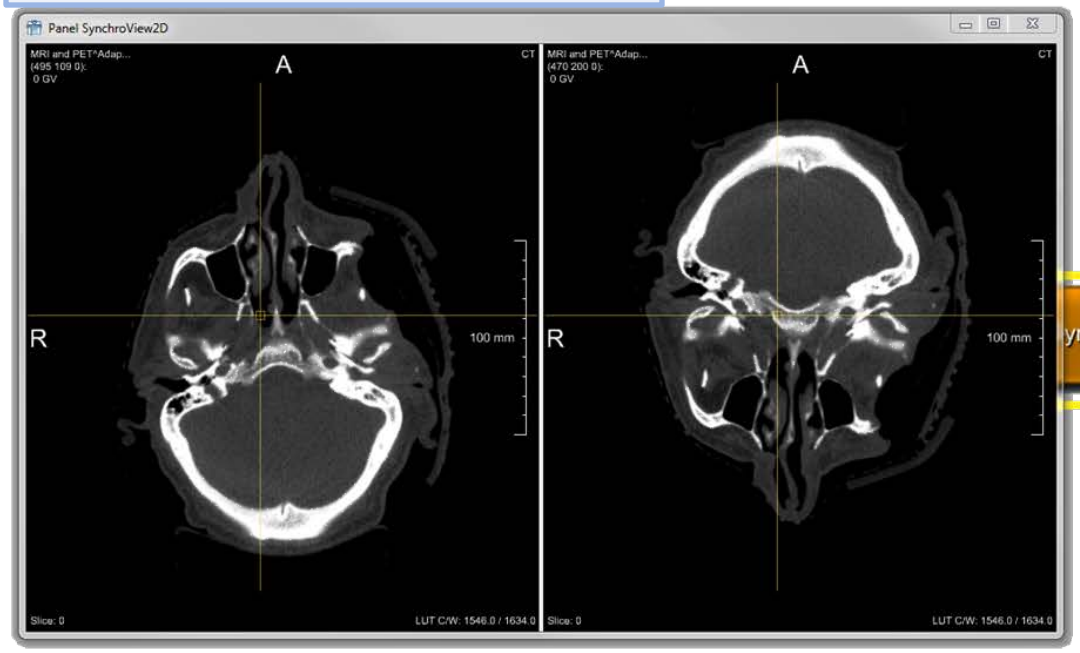




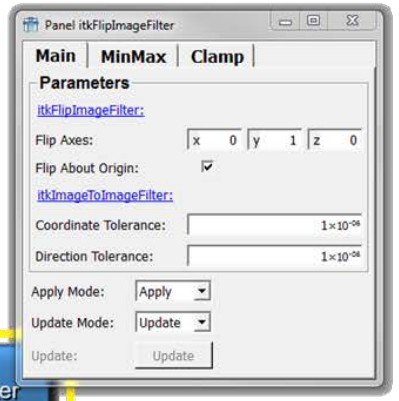
Example

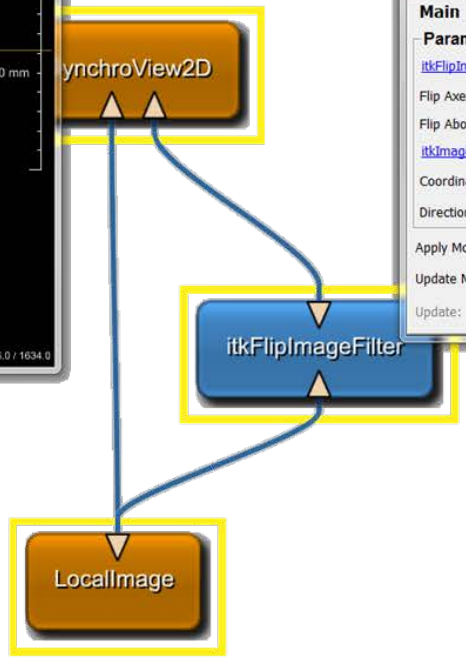
Easy to change setting to experiment on your images

Displays original and processed images

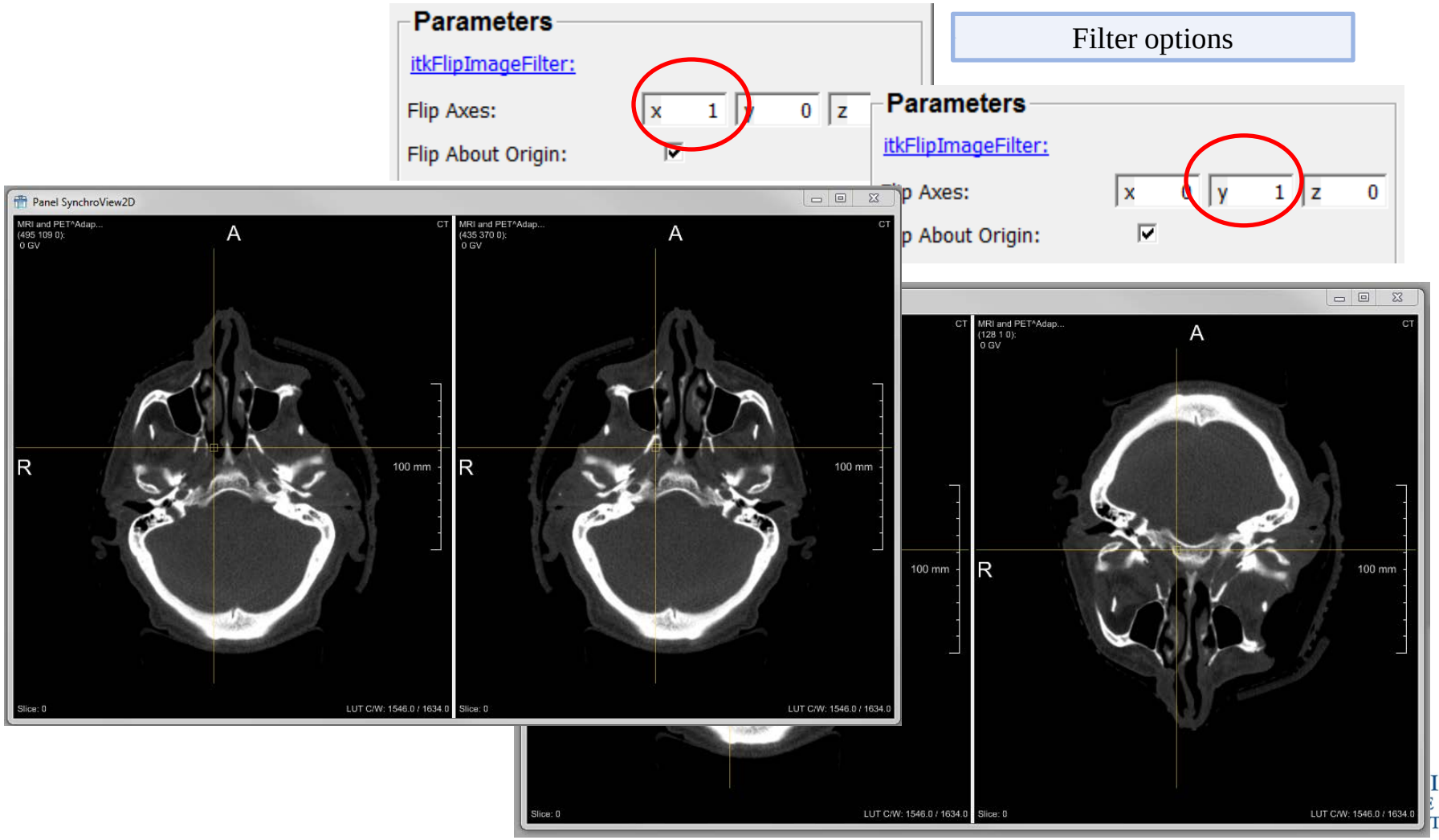


Filter options



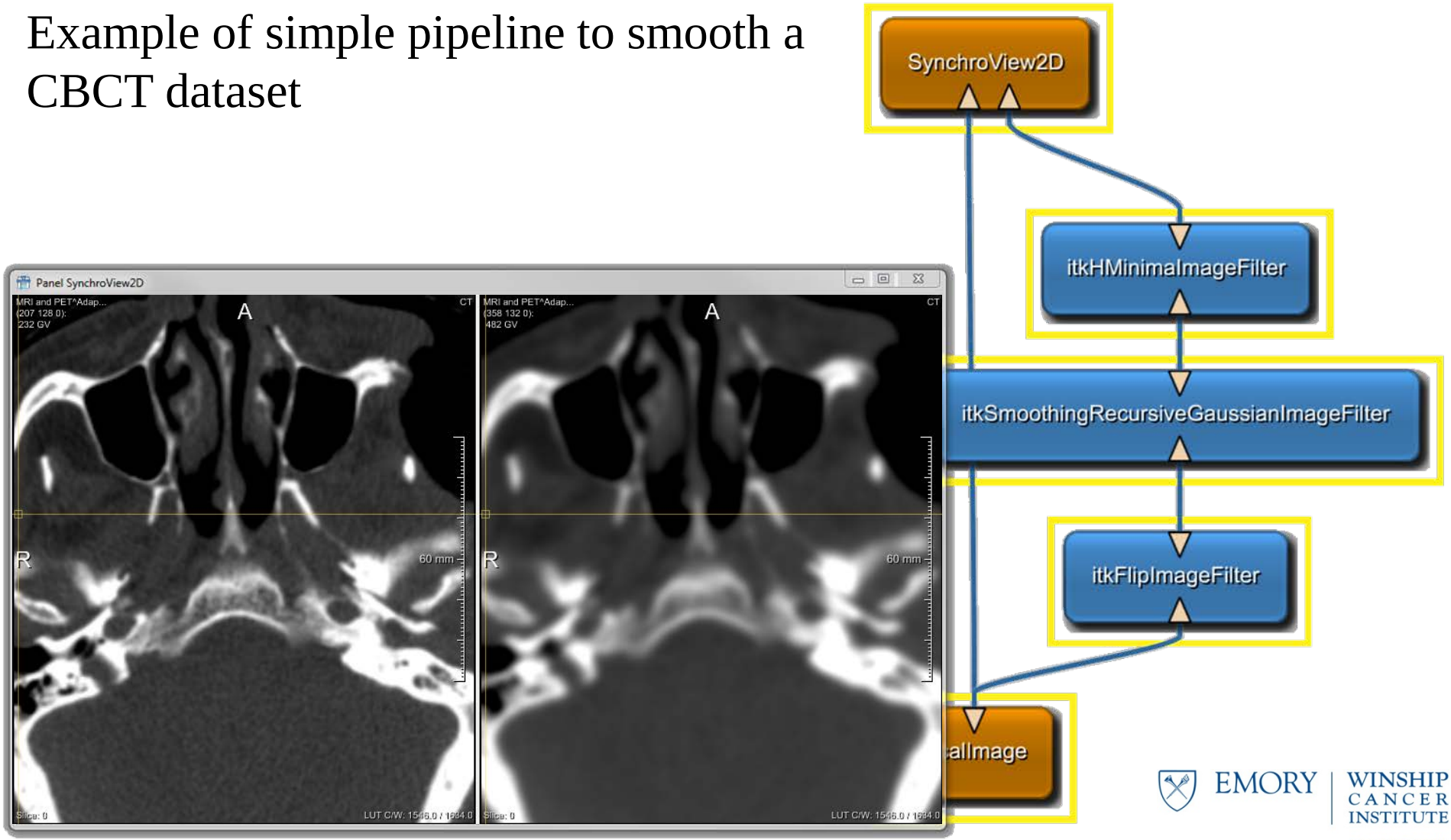


Example



Example

Example of simple pipeline to smooth a CBCT dataset



Example Coding

From prototype to coding

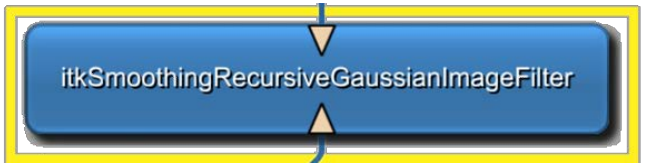
```
int  main (int argc, char *argv[])
{
    //define the image type
    typedef itk::Image< long, 3 >        ImageType;

    //read an image
    typedef itk::ImageFileReader< ImageType > ReaderType;
    ReaderType::Pointer imageReader=  ReaderType::New();
    imageReader->SetFileNames("c:\\myFile.dcm");

    //smoothing filter
    typedef itk::SmoothRecursiveGaussianFilter< ImageType, ImageType > SmoothFilter;
    SmoothFilter::Pointer smoothFilter = SmoothFilter::New();
    smoothFilter->SetInput( imageReader->GetOutput() );
    smoothFilter->Update();



}
```



Template Code

It’s easy to readapt to code to new situations

ITK Definition:

```
template< class TPixel, unsigned int VImageDimension >
Image< TPixel, VImageDimension >
::Image()
{
  m_Buffer = PixelContainer::New();
}
```

Your Code:

```
int  main (int argc, char *argv[])
{

    typedef itk::Image< long, 2 >      SliceType;          // a CT slice

    typedef itk::Image< long, 3 >      VolumeType;         // a CT volume

    typedef itk::Image< long, 4 >      Scan4DType;         // a 4D CT scan

    typedef itk::Image< float, 3 >     DoseType;           // a dose volume
```

Your Own Filter

Deriving a filter from the ITK objects makes my life easier.

Deriving a new inherited filter

```
template< class TImage>
class MyOwnImageFilter:public ImageToImageFilter< TImage, TImage >
{
    public:
```

Implementing your own equation, f(x) = x1.23456

```
inline TOutput operator()(const TInput & A) const
 {
   const double input  = ( static_cast< double >( A )) ;
   const double output = pow(input,1.23546);

   return static_cast< TOutput >( output );
 }
```

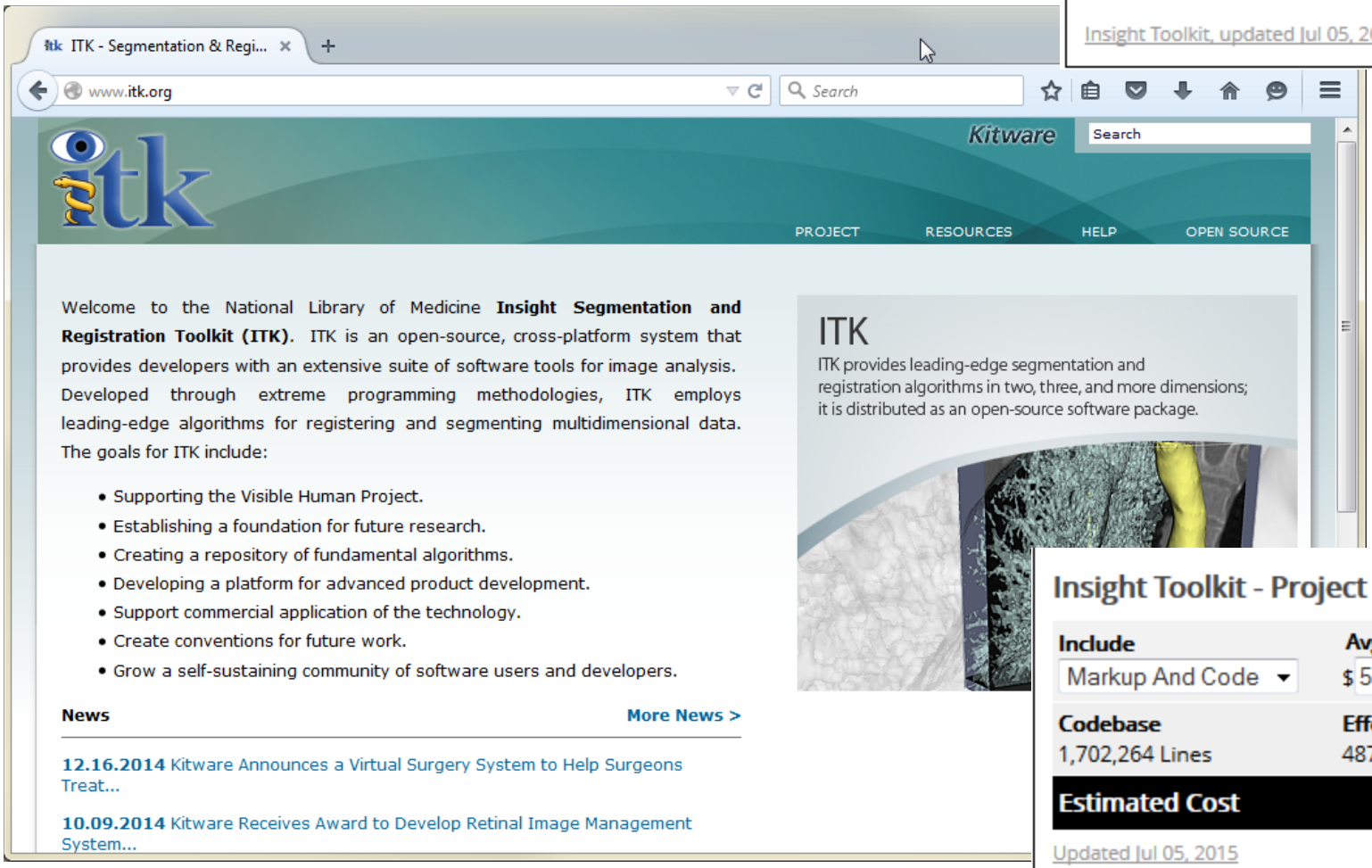
Catalog of ITK Features

- Image IO
- Image processing
  - Canny Edge
  - Hough Transform (lines/ellipsoids)
  - Variable Conductance Diffusion
- Geometry IO/representation/processing (Spatial Objects)
- Statistics
- Registration/Segmentation
- Numerics (VXL)
- Optimizers
- Finite Element Simulation

A Big Library

- Mostly written in C++
- Mature, well-established codebase
- Very large, active development team
- Well-commented source code
- Stable Y-O-Y development activity

Insight Toolkit, updated Jul 05, 2015 more at Open HUB

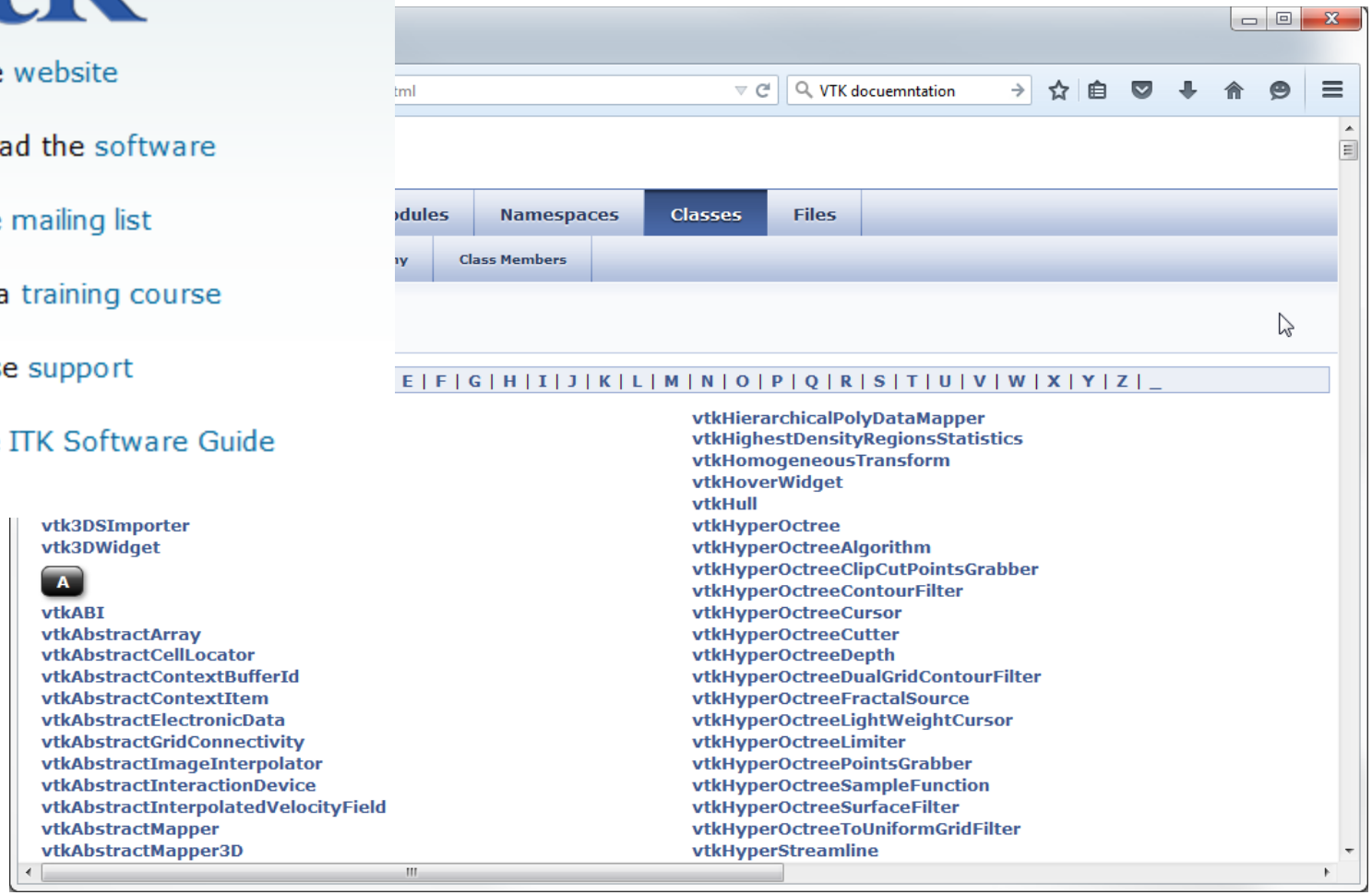


Insight Toolkit - Project Cost

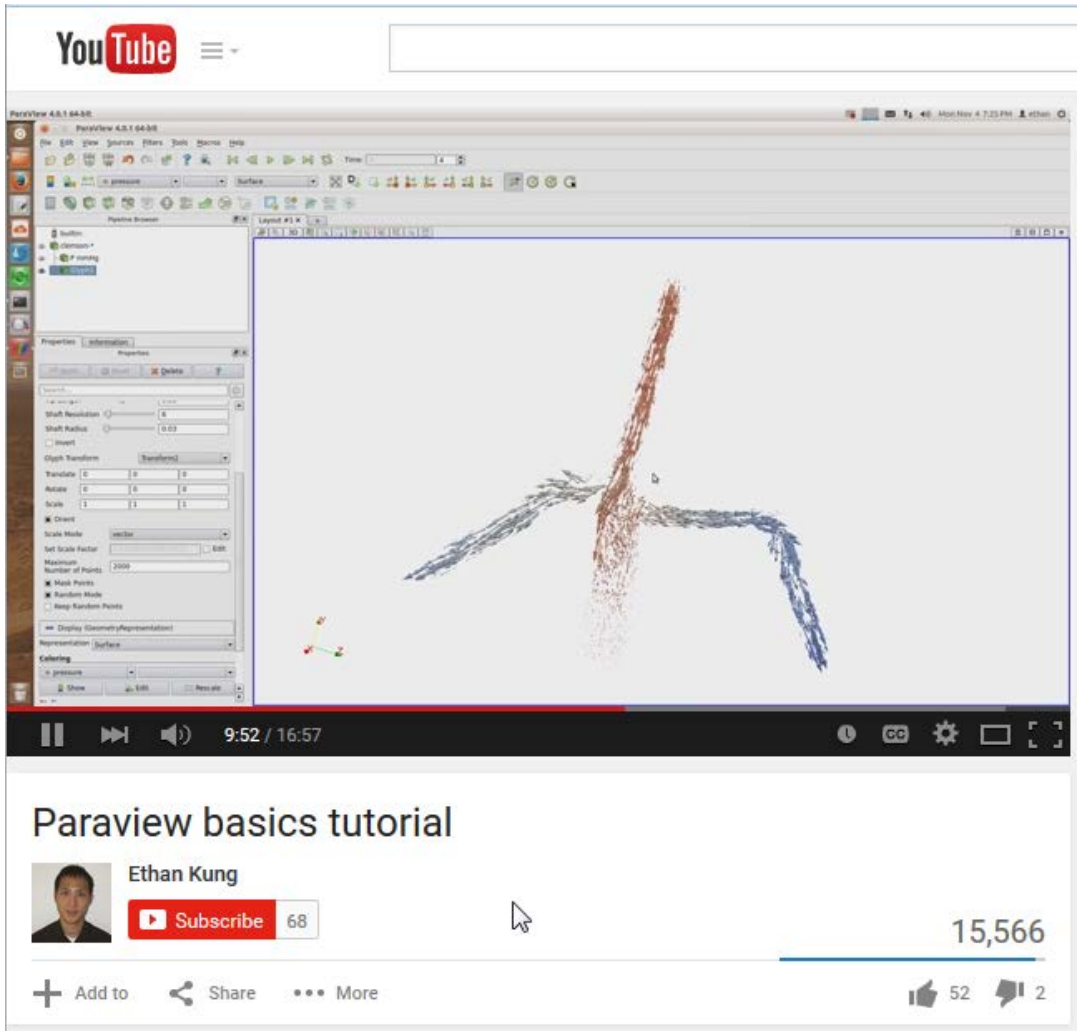
Include	Avg. Salary
Markup And Code	$ 55000 /year
Codebase	**Effort (est.)**
1,702,264 Lines	487 Person Years
Estimated Cost	**$26,785,280**

Updated Jul 05, 2015 more at Open HUB

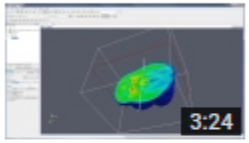
Where To Find Help



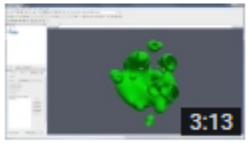
Lots of Documentation



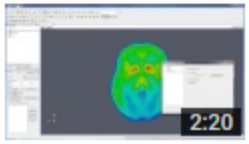
[ParaView] Basics of Keyframe Animation
by UM3DLab
6,277 views



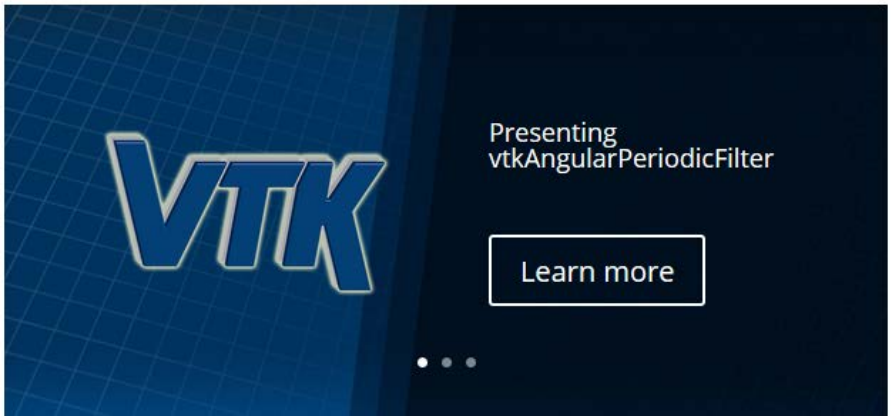
[ParaView] Basics of Clipping
by UM3DLab
4,098 views

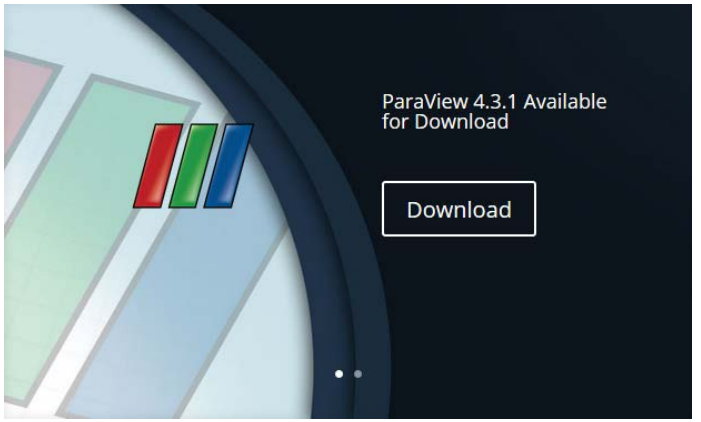


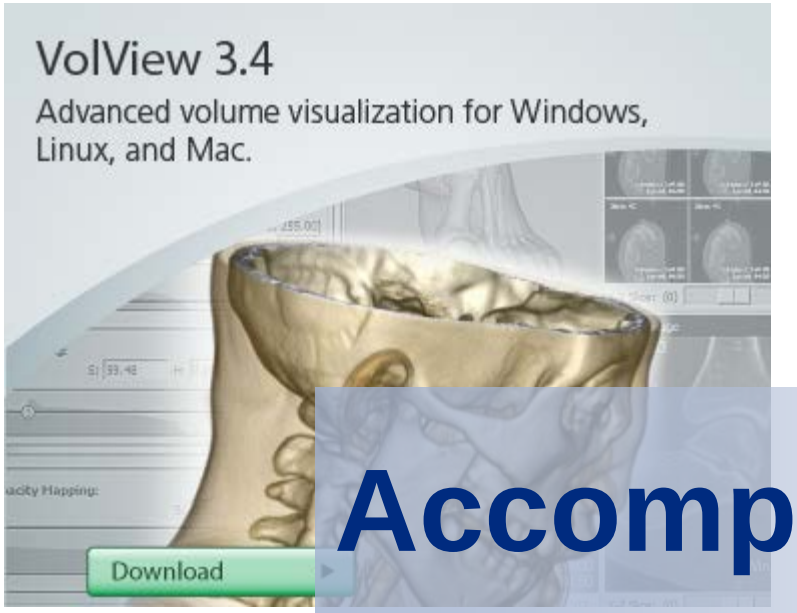
[ParaView] Creating IsoSurfaces
by UM3DLab
5,544 views



[ParaView] Basics of Slicing and Planar Plots
by UM3DLab
4,329 views

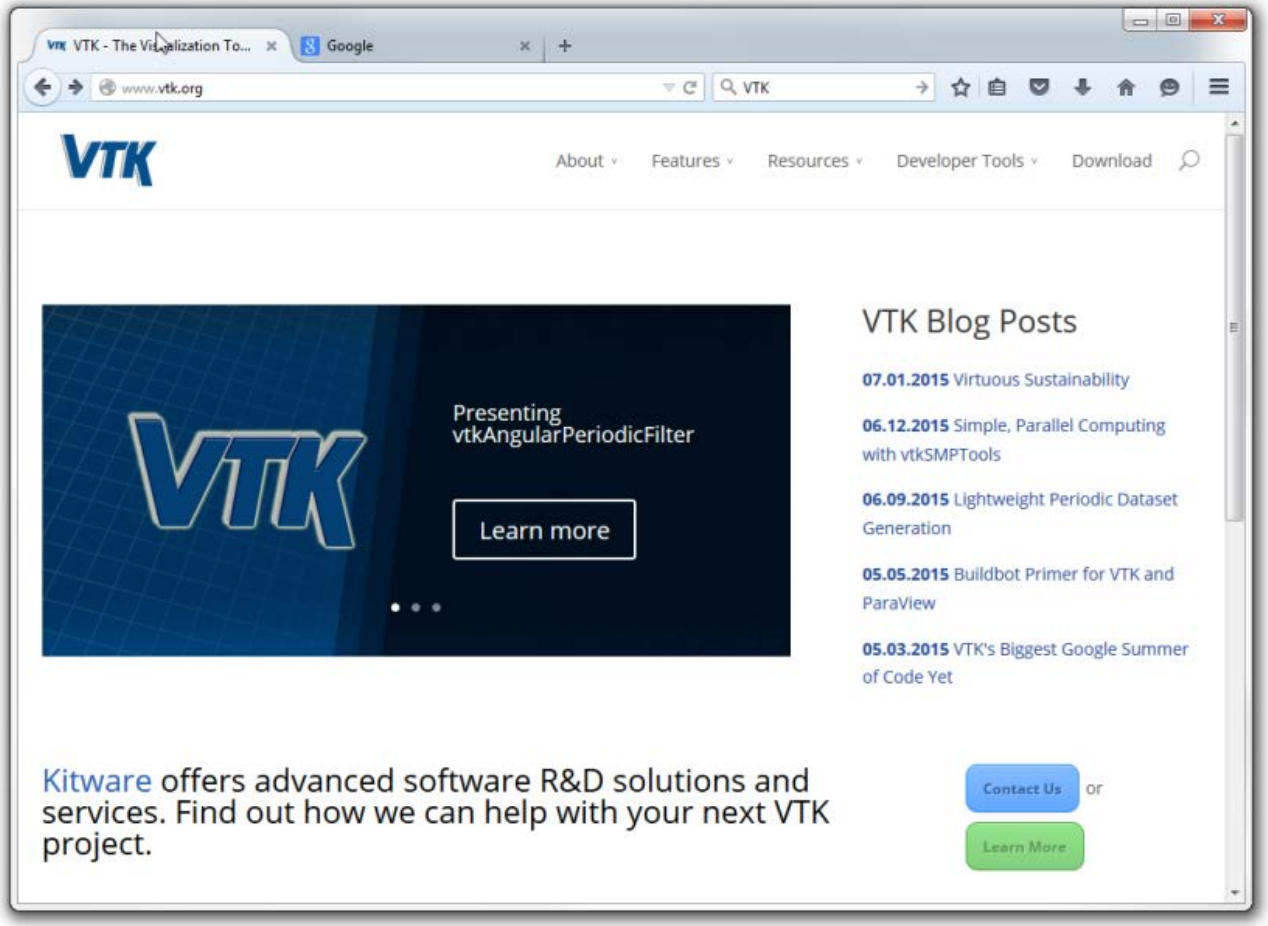






Accompanying Libraries

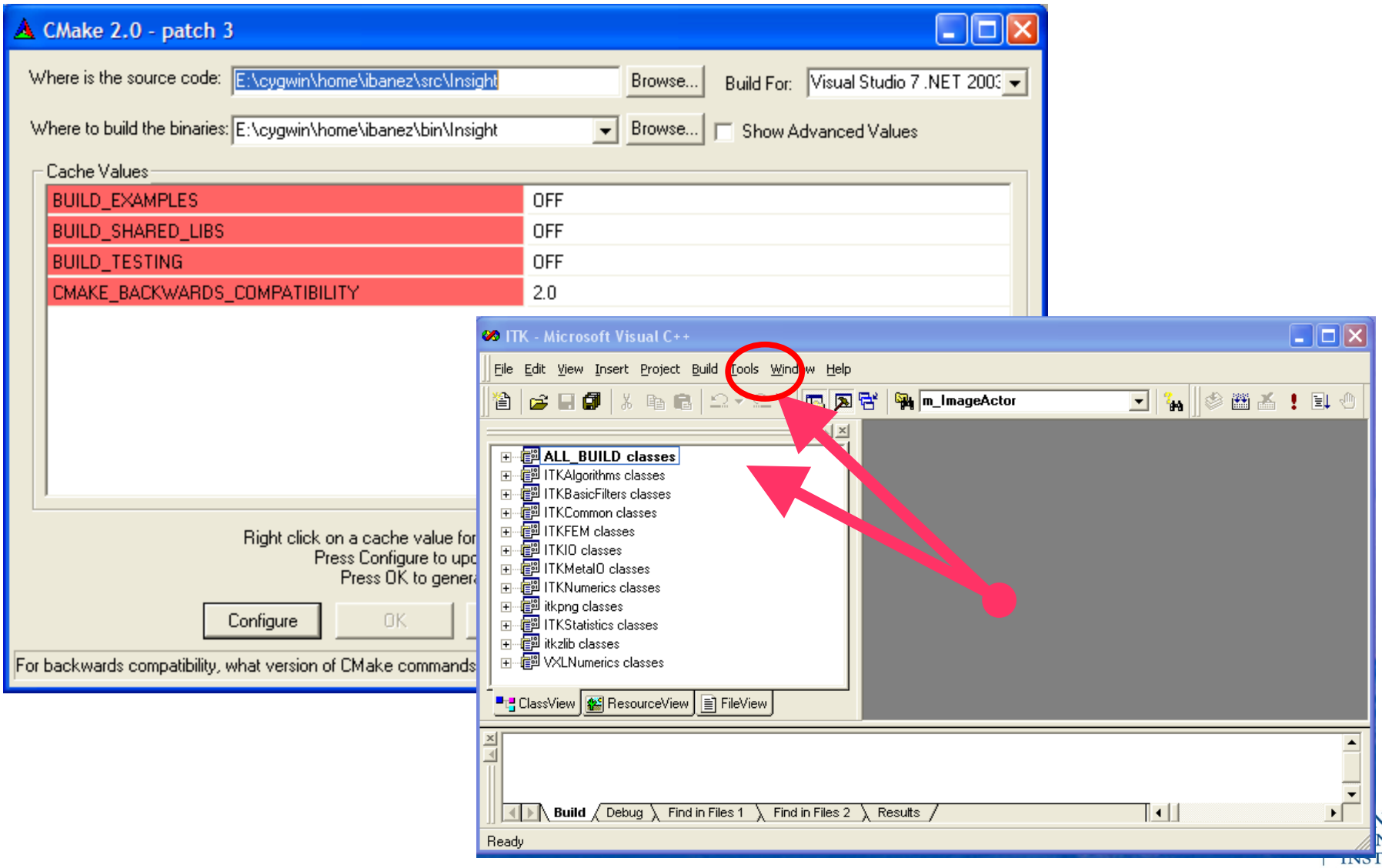
Visualization Tool Kit



Features ▾ Resources ▾ D

- Open Source
- Platform Agnostic
- Language Agnostic
- Data Model
- Visualization
- Modelling
- Imaging
- 3D Graphics
- 2D Plots and Charts
- Interaction
- InfoVis
- Parallel Processing

CMake



Where to start

MeVisLab

www.mevislab.de

www.itk.org

www.volview.org

www.itk.org

www.paraview.org

www.cmake.org