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.
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.
My Toolbox

- **MeVisLab**: Rapid prototyping
- **itk**: Repository of image processing algorithms
- **VolView**: Visualize images
- **VTK**: Repository of visualization algorithms
- **ParaView**: Visualize meshes
- **CMake**: 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.
Prototyping

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.

- A filter (building block) that does some processing to an image
- A connection between the building blocks
The `itkFlipImageFilter` will flip an image along a user specified axis.

- **SynchroView2D**
- **itkFlipImageFilter**
- **LocalImage**
- **Displays original and processed images**
- **Process image: flips along one axis**
- **Reads an image from disk**
Example

Easy to change setting to experiment on your images

Displays original and processed images
Example
Example of simple pipeline to smooth a CBCT dataset
Example Coding

From prototype to coding

```c
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:**

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

**Your Code:**

```cpp
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

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

Implementing your own equation, \( f(x) = x^{1.23456} \)

```cpp
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
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.

Insight Toolkit - Project Cost

<table>
<thead>
<tr>
<th>Include</th>
<th>Avg. Salary</th>
<th>Effort (est.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markup And Code</td>
<td>$55000</td>
<td>487 Person Years</td>
</tr>
<tr>
<td>Codebase</td>
<td>1,702,264 Lines</td>
<td></td>
</tr>
</tbody>
</table>

Estimated Cost: $26,785,280
Where To Find Help

- Visit the website
- Download the software
- Join the mailing list
- Attend a training course
- Purchase support
- Buy the ITK Software Guide
Lots of Documentation
VolView 3.4
Advanced volume visualization for Windows, Linux, and Mac.

Accompanying Libraries
Visualization Tool Kit

Kitware offers advanced software R&D solutions and services. Find out how we can help with your next VTK project.
CMake
Where to start

www.mevislab.de

www.volview.org

www.paraview.org

www.itk.org

www.itk.org

www.cmake.org