SimpleITK Status

Bill Hoffman, Jesus Caban, Brad Lowkamp, Dan Muller, Fabrice de Chaumont, Julien Michel, Harvey Cline, Gabe Hart, Ghassam Hamarech, John Galeotti, Raghu Machiraju, Hans Johnson, Wes Turner, Luis Ibanez, Terry Yoo, Daniel Blezek

MAYO CLINIC

Today’s Plan

```
$ cat .plan
Build a better mousetrap
Rule world
```

- Discuss viewpoints
- Survey
- Coffee cup
- Decide
- Go home and code

Problem

```
// Setup image types.
typedef    float    InputPixelType;
typedef    float    OutputPixelType;

typedef itk::Image< InputPixelType,  2 >   InputImageType;
typedef itk::Image< OutputPixelType, 2 >   OutputImageType;

// Filter type
typedef itk::DiscreteGaussianImageFilter<
              InputImageType, OutputImageType >  FilterType;

// Create a filter
FilterType::Pointer filter = FilterType::New();

// Create the pipeline
filter->SetInput( reader->GetOutput() );
filter->SetVariance( 1.0 );
filter->SetMaximumKernelWidth( 5 );
filter->Update();
OutputImageType::Pointer blurred = filter->GetOutput();
```

Listing 1: Discrete gaussian filter implemented in ITK

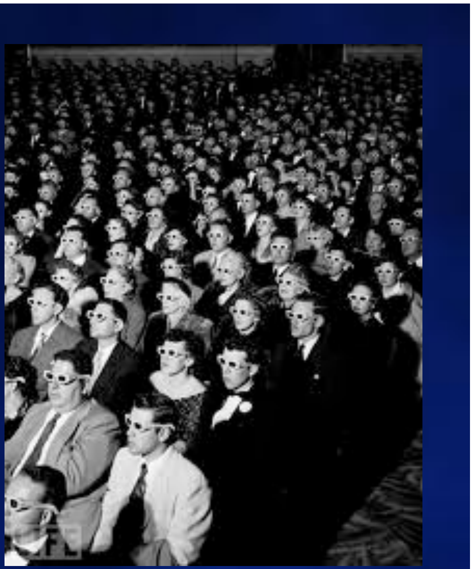
```
% Filter image I with a gaussian of sigma 1 with a radius 5 kernel.
blurred = imfilter ( I, fspecial ( 'gaussian', 5, 1 ) );
```

Listing 2: Discrete gaussian filter implemented in Matlab

Complexity

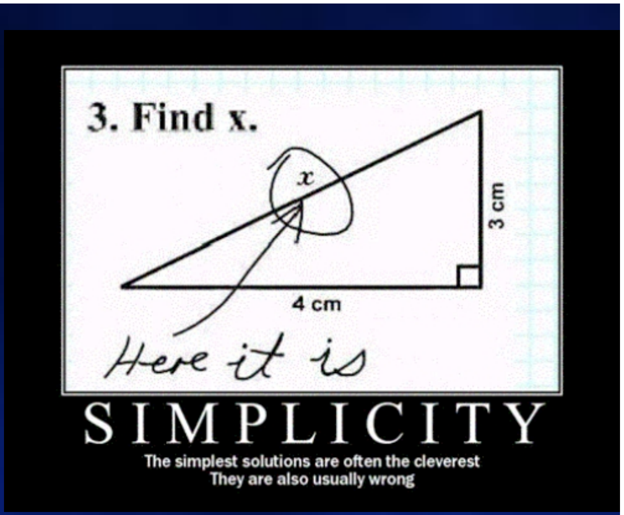
- Powerful but complex
- Steep learning curve
  - ever had 2 pages of compile errors?
- C++
- ITK often unused
  - But it’s the best...

Audience



- Students
- Grad students
- Researchers
  - Microscopy
  - Computer vision
  - Remote sensing
- Application developers

Simplify


3. Find x.
x
3 cm
4 cm
Here it is
SIMPLICITY
The simplest solutions are often the cleverest
They are also usually wrong

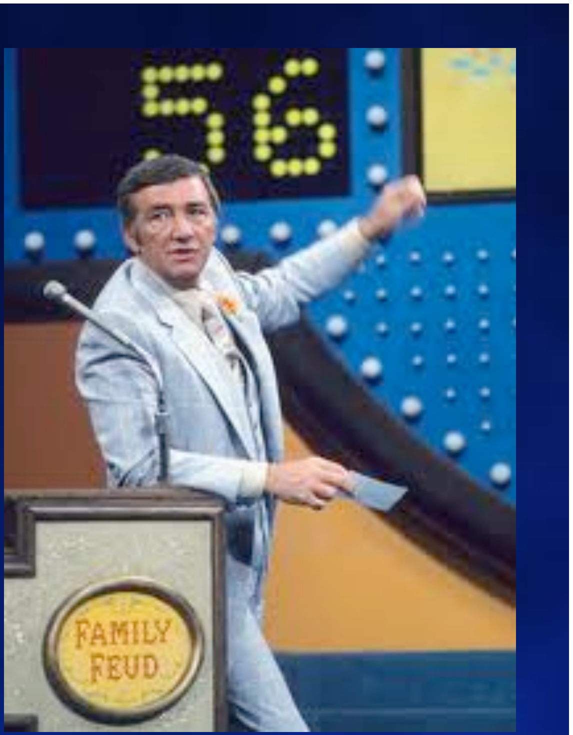
- As simple as possible...
- ...but no simpler
- Simple for:
  - users
  - application developers
  - SimpleITK developers

Goals

- Simple
- “On ramp”
- Wrapping
- Integration
  - In apps
  - With ITK, VTK, etc
- Extensible

Survey Says!

- Compiled packages
- Documentation / Examples
- 3D & 4D
- Out of core processing
- Object model
- char, short, float/double, vector(?)
- Visualization



Gabe Hart

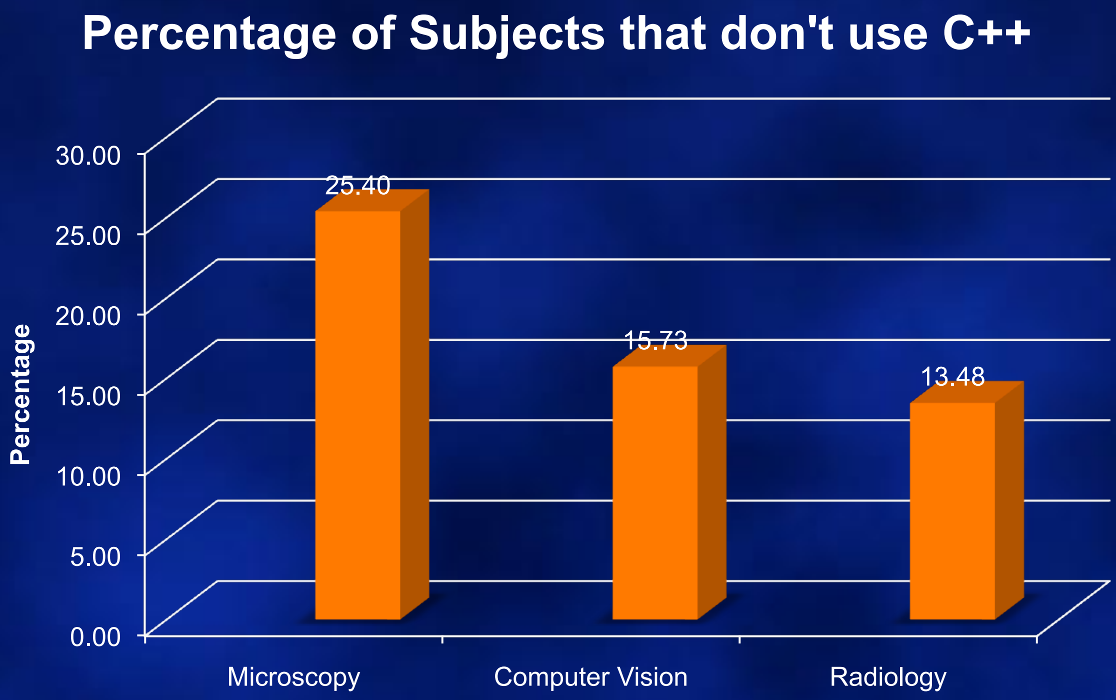
Getting inside their heads

- 253 participants (11/6/10)

- Non-C++ users:
  - 21.34% (54 participants) have never used or don't feel comfortable using C++
    - Specific distribution: 16 + 38
  - 51% (28) of non-C++ users have used ITK
  - 26 have never used ITK

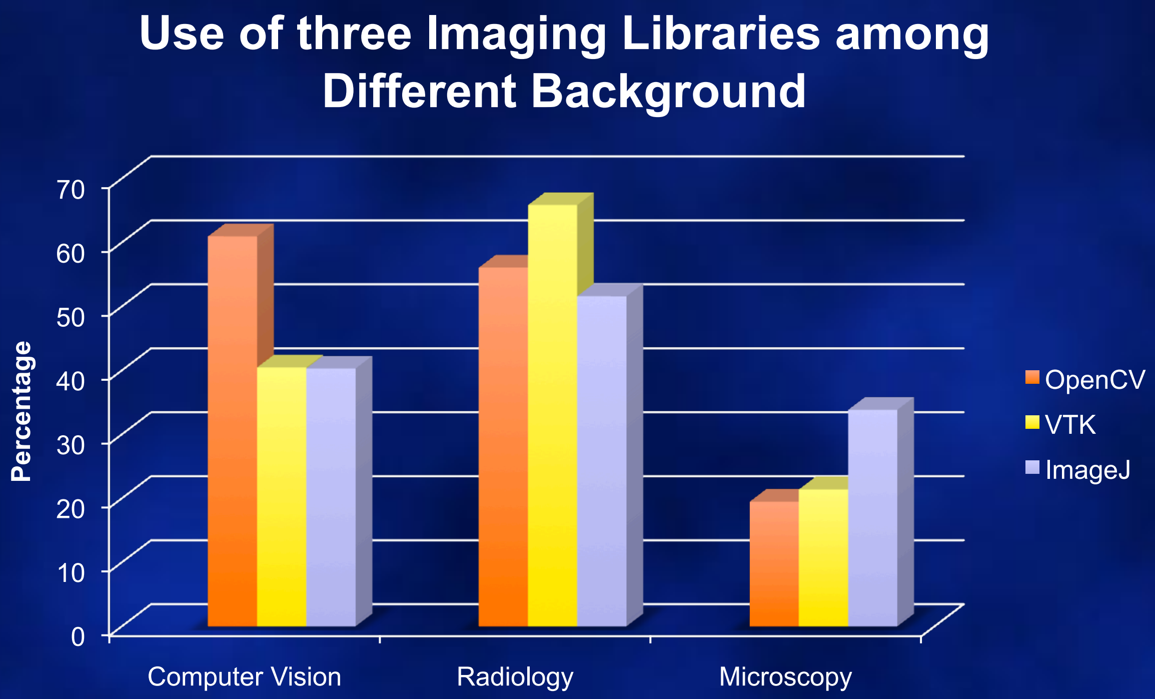
- Non-ITK users:
  - 17.78% (45 participants) have never used ITK
  - 207 have used ITK
  - 44.44% (20) of non-ITK users are C++ users

C++



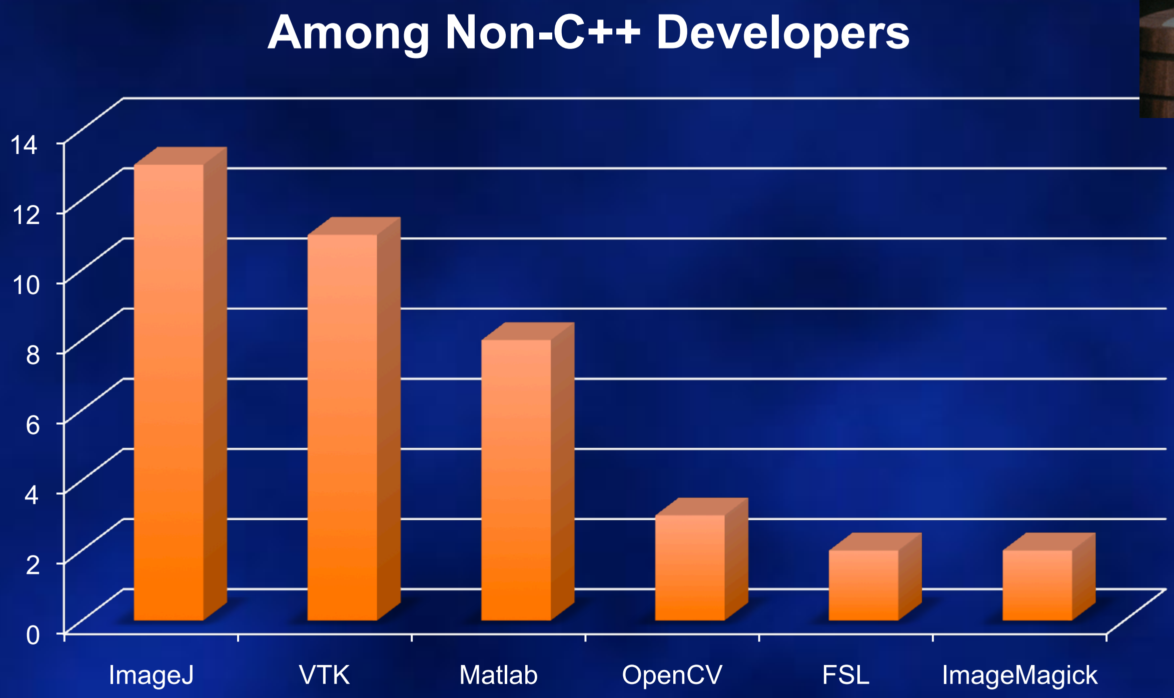
MAYO CLINIC

Competition



MAYO CLINIC

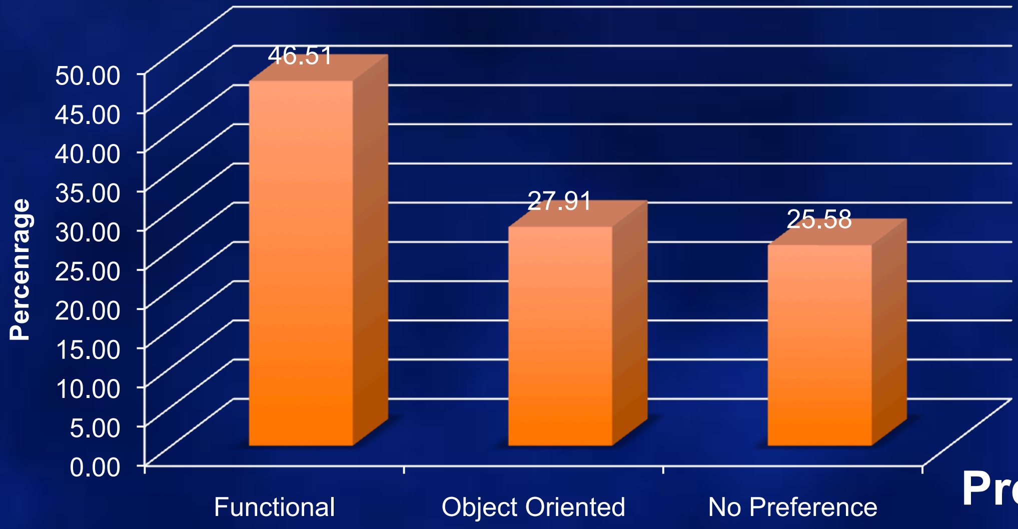
Competition



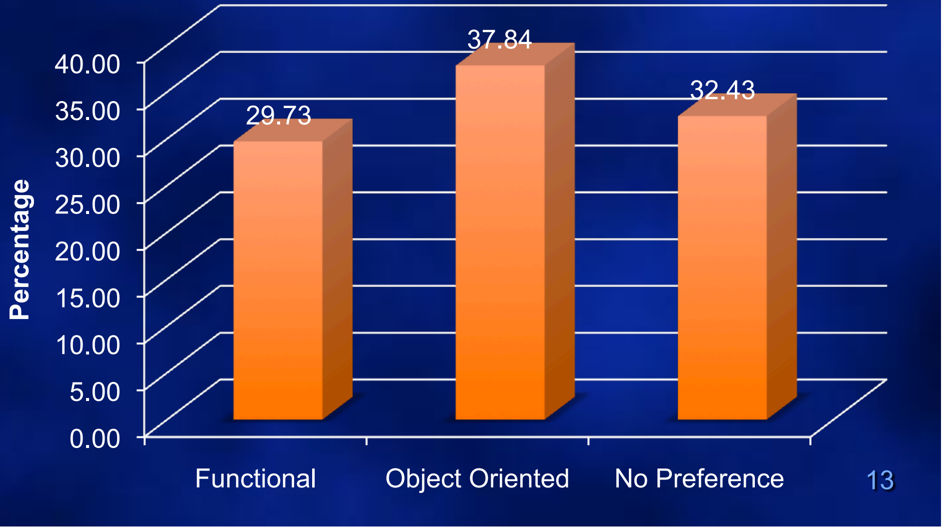
Note that all of the top libraries / applications among non-C++ developers provide processing and visualization.

Functional / Objects

Preference Among Non-C++ Programmers

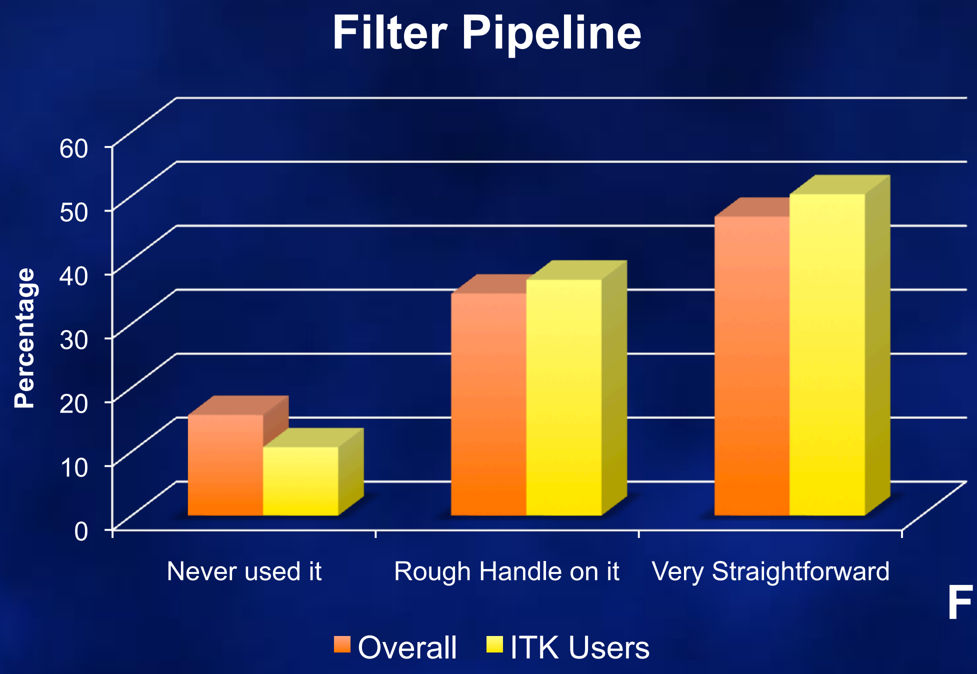


Preference Among Non-ITK Users



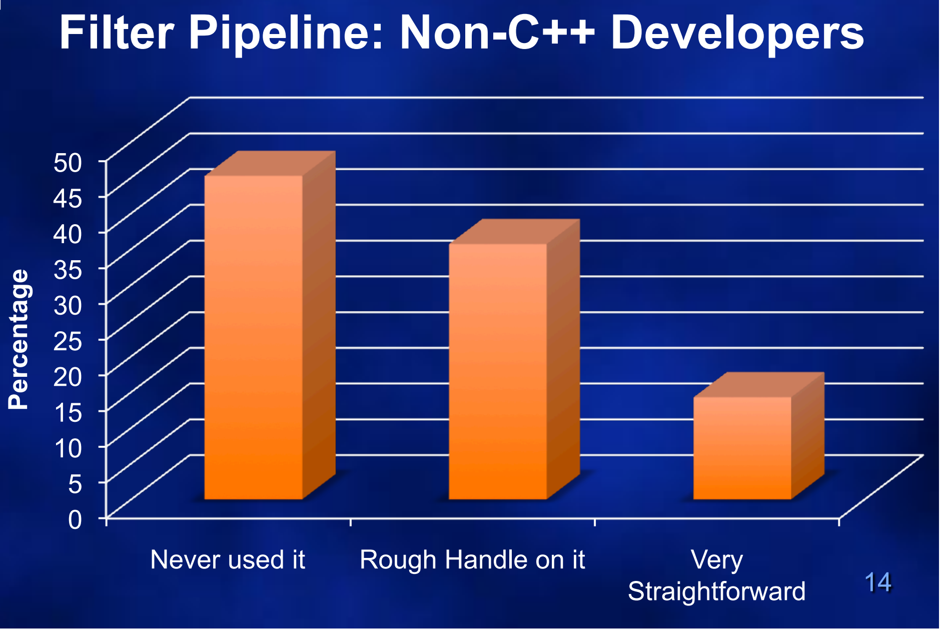
MAYO CLINIC

Pipeline

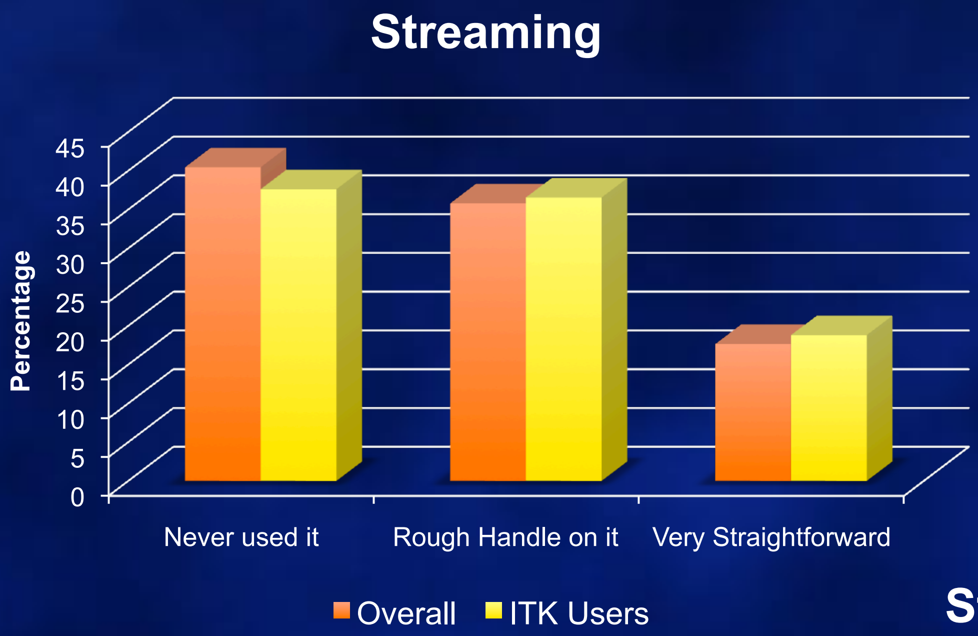


Note:

Most non-C++ users (including the 51% that have used ITK), don't understand the pipeline.

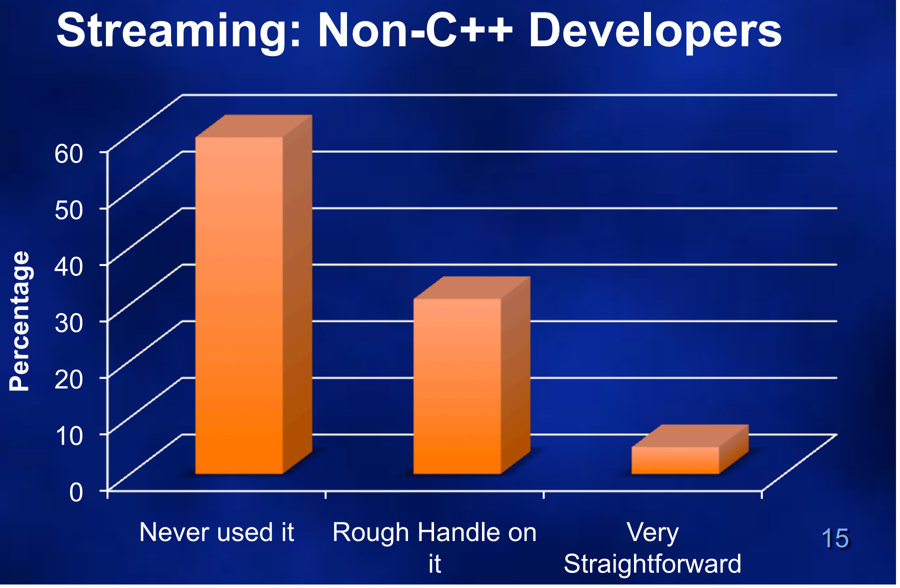


Streaming



Note:

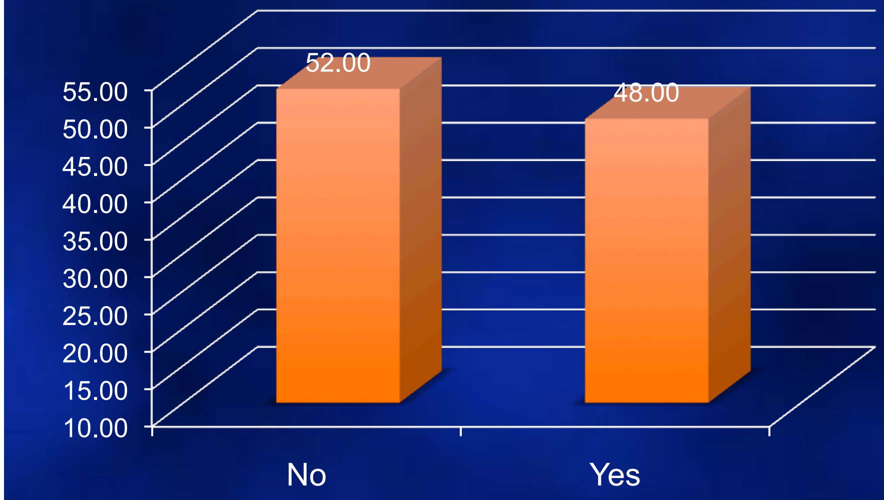
Most ITK, non-ITK, and C++ users don't have a clear understanding of streaming



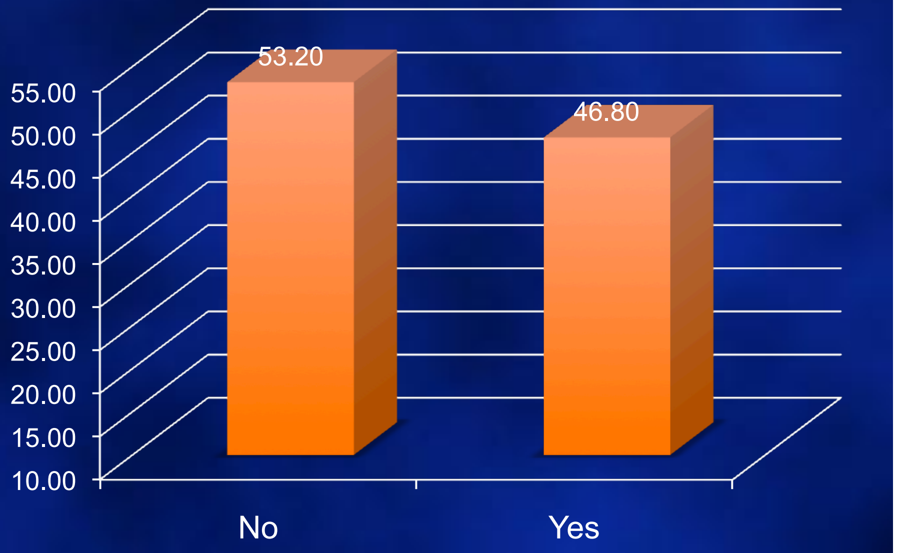
MAYO CLINIC

Out-of-core

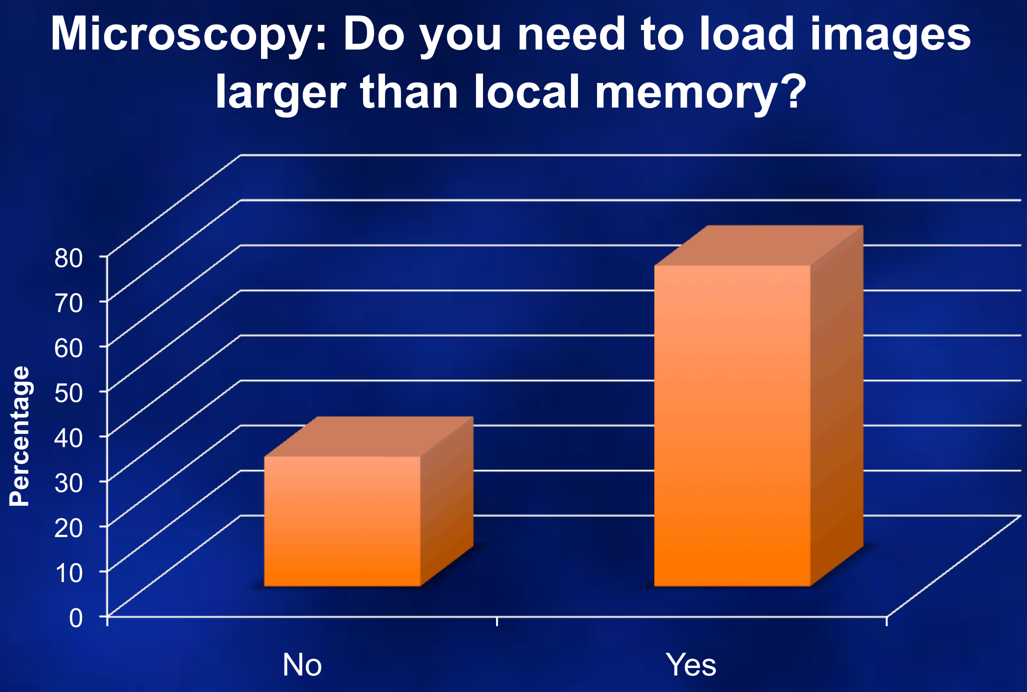
Non-C++ Users: Do you need to load images larger than local memory?



ITK Users: Do you need to load images larger than local memory?



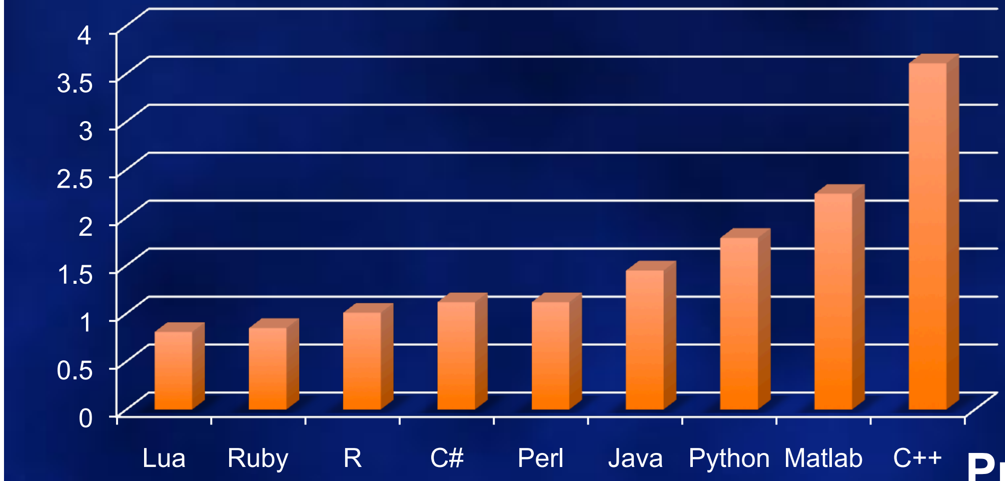
Out-of-core


Microscopy: Do you need to load images larger than local memory?
Percentage
80
70
60
50
40
30
20
10
0
No
Yes

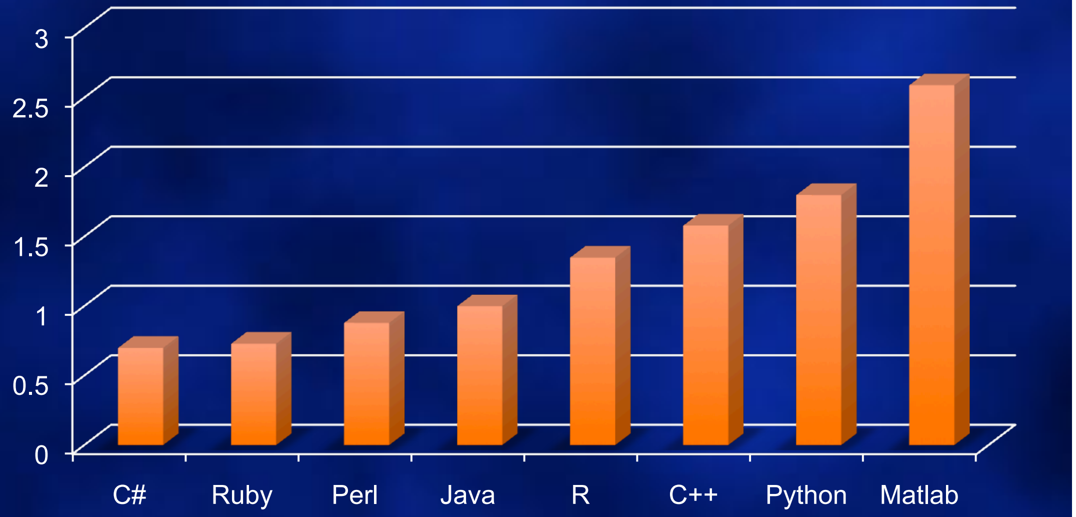
MAYO CLINIC

Languages

Preferences: C++ Experts



Preferences: Non C++ Developers



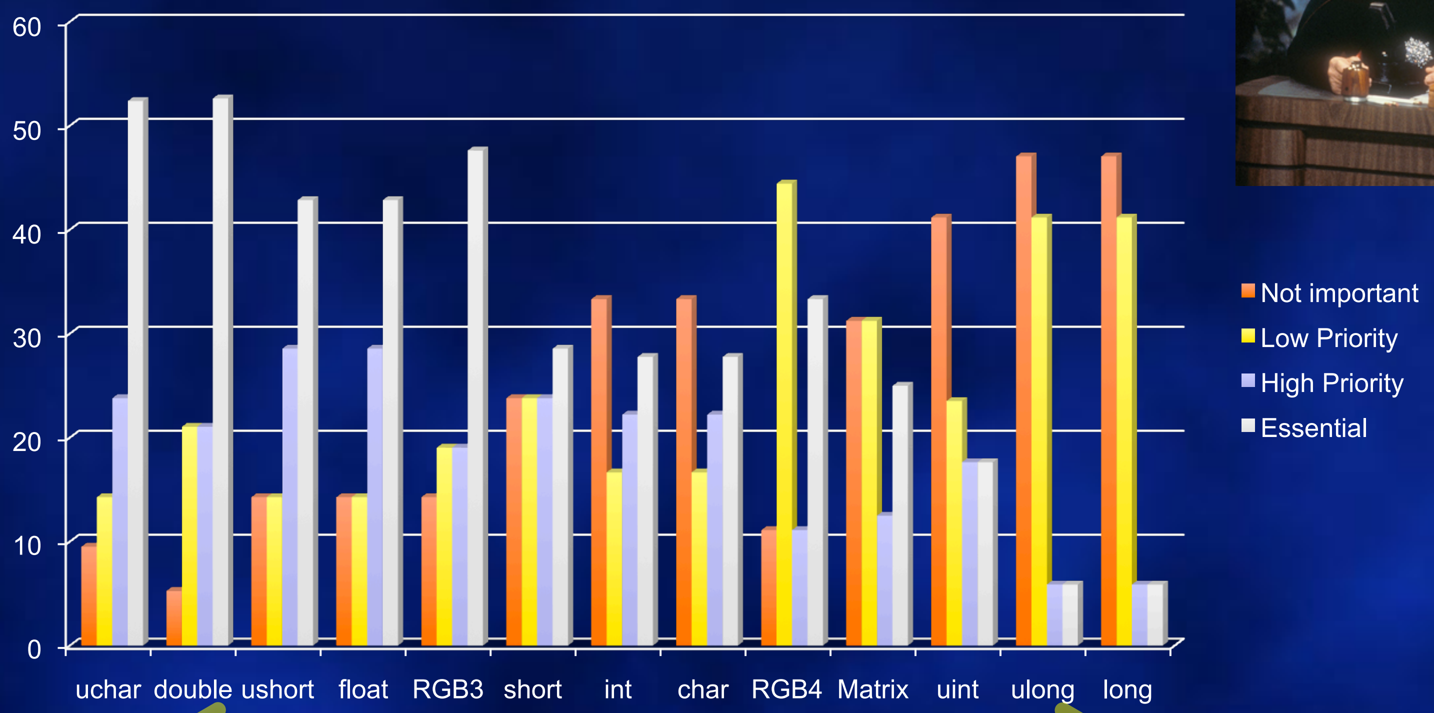
Q:

Can we say that Matlab, Python, C++, Java, and R are the most important programming languages?

MAYO CLINIC

Pixel Types

Pixel Type: Non-ITK Users



More important ⟷ Less Important

Top 6 (Non-ITK users)

1. uchar
2. double
3. ushort
4. float
5. RGB3
6. short

Top 6 (Overall)

1. float
2. uchar
3. ushort
4. short
5. double
6. matrix

Comments: defaults

- "Important to give users relatively straightforward advice on algorithm…"
- "Simple defaults with optional alternatives"
- "I generally want to be able to try something off the shelf with sensible defaults"
- "Sensible defaults should be provided to avoid long code"
- "Full flexibility is needed but a good selection of default parameters for all modules would be also important"
- "initialized with reasonable default components"

Comments: GUI

- “offer both...a GUI … and build scripts from there”
- “connect display to view intermediate images”
- “handling ability to visualize quickly (even at run time) images and histograms”
- “A GUI to build up the application pipeline by arranging icons of filters and I/O connectors”
- “visual tool for connecting filters to a pipeline

Comments: Complexity

- “writing ITK codes can be a complex task but with more examples and a thorough documentation these difficulties can be overcame”
- “One good way to compromise is more flexible but with lots of example code that can be cut-and-pasted for typical applications.”
- “Actually I think much of the motivation for SimpleITK could go away, if someone updated + improved the documentation”
- “some of the image filters modules are not sufficiently documented”
- “Lots of sample code that can be modified for specific purposes”

Ouch

- “I am not using ITK for registration purposes, and this is because of the complexity of it”
- “Extensive use of templates makes debugging difficult in VisualC++”
- “Most of filter called *VectorImageFilter are in fact not able to process itk::VectorImage ... This is confusing and sometimes frustrating”
- Ability to open differing file formats (dimensions, data types) without all the stupid c++ template nonsense.

Decisions

- API
- Datatypes
- Dimensionality
- Pipeline?
- Out of core processing
- Wrapping
- Distribution

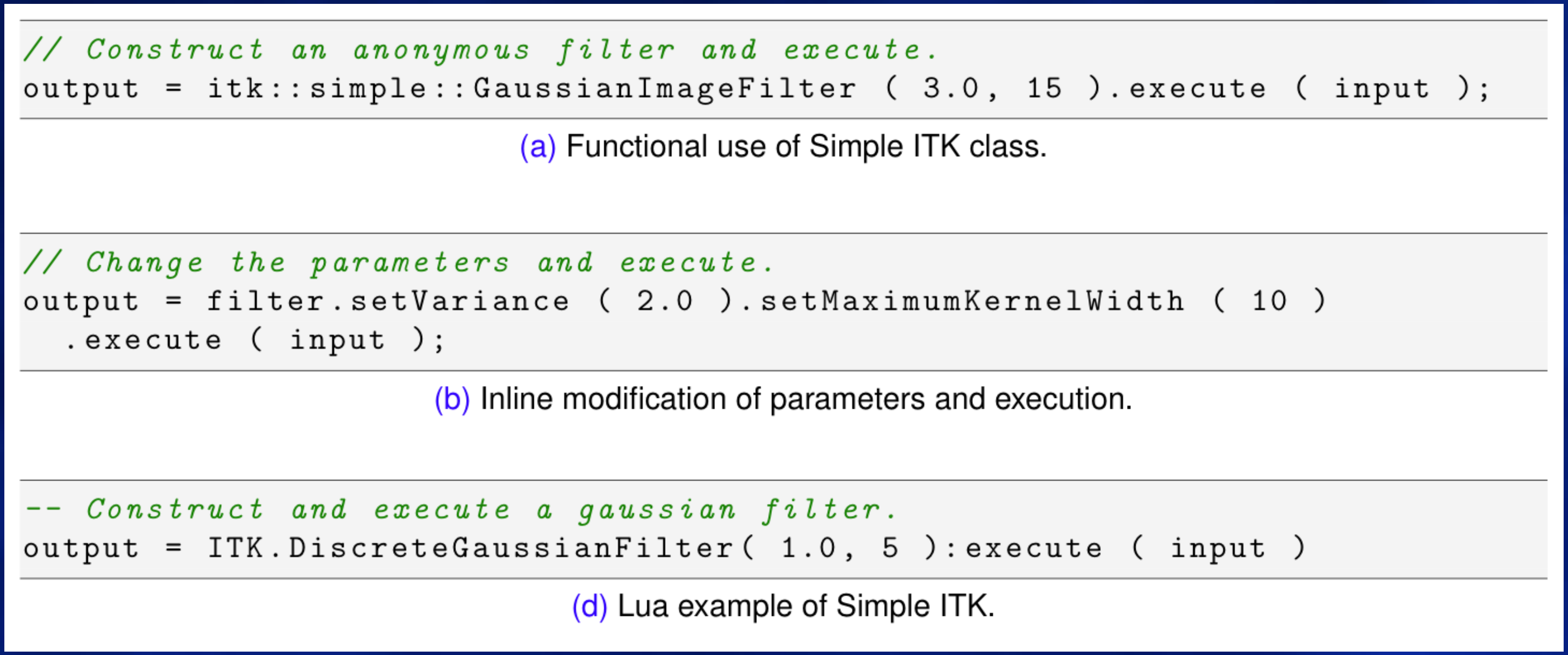
Coffee-cupping

Strawman Proposal

- Object model
- 3D images (& slice-by-slice for 2D)
  - 4D as vector of images, labelmap class
  - char, short, float (double?)
- No exposed pipeline
- Out of core orchestration framework
  - VOI readers/writers
  - Call list of filters / handle boundaries
- SWIG wrapping

API

- **Procedural / functional?**



```
// Construct an anonymous filter and execute.
output = itk::simple::GaussianImageFilter ( 3.0, 15 ).execute ( input );
```

(a) Functional use of Simple ITK class.

```
// Change the parameters and execute.
output = filter.setVariance ( 2.0 ).setMaximumKernelWidth ( 10 )
  .execute ( input );
```

(b) Inline modification of parameters and execution.

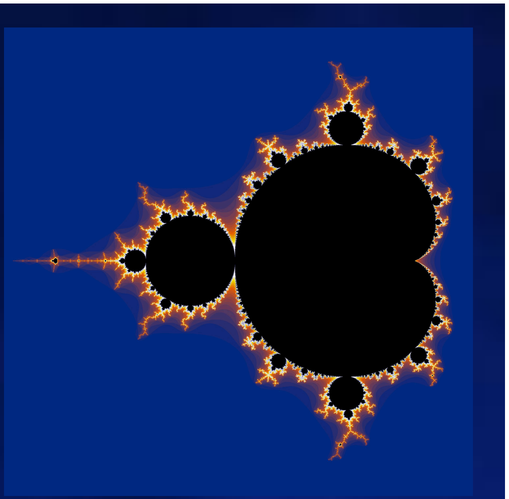
```
-- Construct and execute a gaussian filter.
output = ITK.DiscreteGaussianFilter( 1.0, 5 ):execute ( input )
```

(d) Lua example of Simple ITK.

Alternative

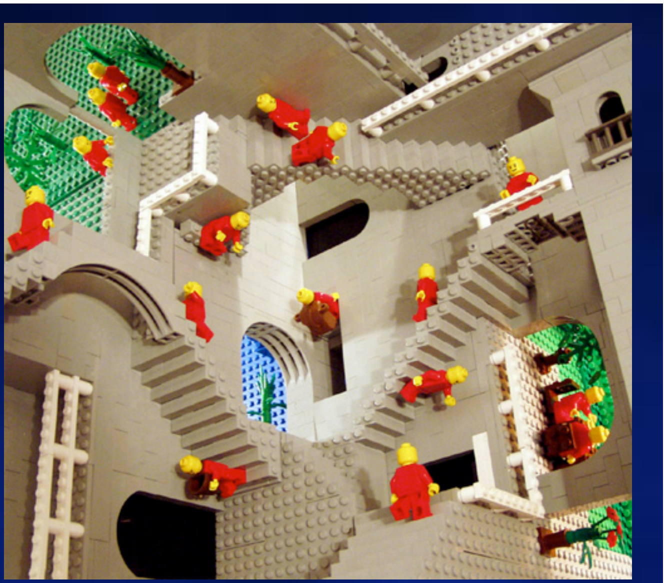
- C++ should be the same as wrapped languages
- Accessibility to as many developers as possible
  - Python, Java, and non-templated C++ programmers
- VTK wraps into python, java, C#, Tcl
  - The C++ code is 90% available
- As VTK grows the simple VTK grows
- What is different than ITK?
  - Templates!
- Custom API would get old and not maintained
- For this to “stick” we would have to drink our own beer

Data types



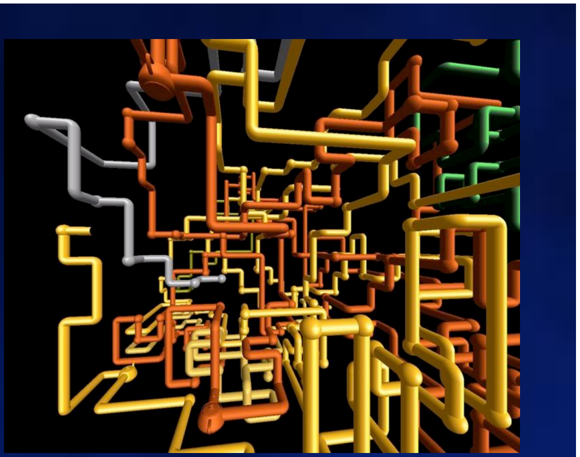
- Which to support?
  - Scalar in all filters?
  - Vector in limited set?
- How to support?
  - Dynamic loading?
  - Compile time configured?
  - Template meta programming?
  - Macros?

Dimensionality



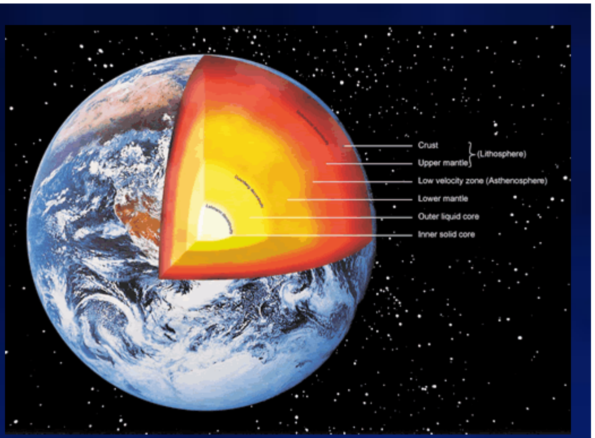
- 3D critical
- 4D nice
- How to support?
    - 4D == 4D itk::Image?
    - 4D == vector of 3D images?
    - 4D == 3D with vector pixels?

Pipeline



- Unnecessary complexity
- Utility in streaming
  - Not (fully) implemented in ITK

Out of core processing



- Essential
  - ITK Pipeline?
  - OpenCL model?
    - single pixel kernel pixel
    - called for all pixels
    - “orchestration” framework

Wrapping

- SWIG
- Languages
  - Matlab
  - Python
  - Java
  - C#
  - Others: Tcl, R, Perl, Ruby...
- Interface to WrapITK a “nice to have”

Distribution

- Binary
  - Matlab (somehow)
  - Python egg
  - Java jar
  - C++ lib/dll/so & include
- Source
  - Less useful

SimpleITK Status

Bill Hoffman, Jesus Caban, Brad Lowkamp, Dan Muller, Fabrice de Chaumont, Julien Michel, Harvey Cline, Gabe Hart, Ghassam Hamarech, John Galeotti, Raghu Machiraju, Hans Johnson, Wes Turner, Luis Ibanez, Terry Yoo, Daniel Blezek