ITKv4 - The Next Generation

What is new in ITKv4

Insight Software Consortium

May 2012

Virtual Machine Preparation I

Virtual Machines Preparation

- Get DVD / USB Memory Stick
- Install VirtualBox from it
- Import the VirtualMachine file
- Boot the Virtual Machine
- Log in
- Get familiar with directories

- VirtualBoxInstallers
  - VirtualBox-4.1.12-77245-OSX.dmg (Mac)
  - VirtualBox-4.1.12-77245-Win.exe (Windows)

Media Content

Directories and Files

- VirtualBoxInstallers
  - VirtualBox-4.1.12-77245-OSX.dmg (Mac)
  - VirtualBox-4.1.12-77245-Win.exe (Windows)
- VirtualMachine
  - ITK-Barcelona.ova

Install VirtualBox

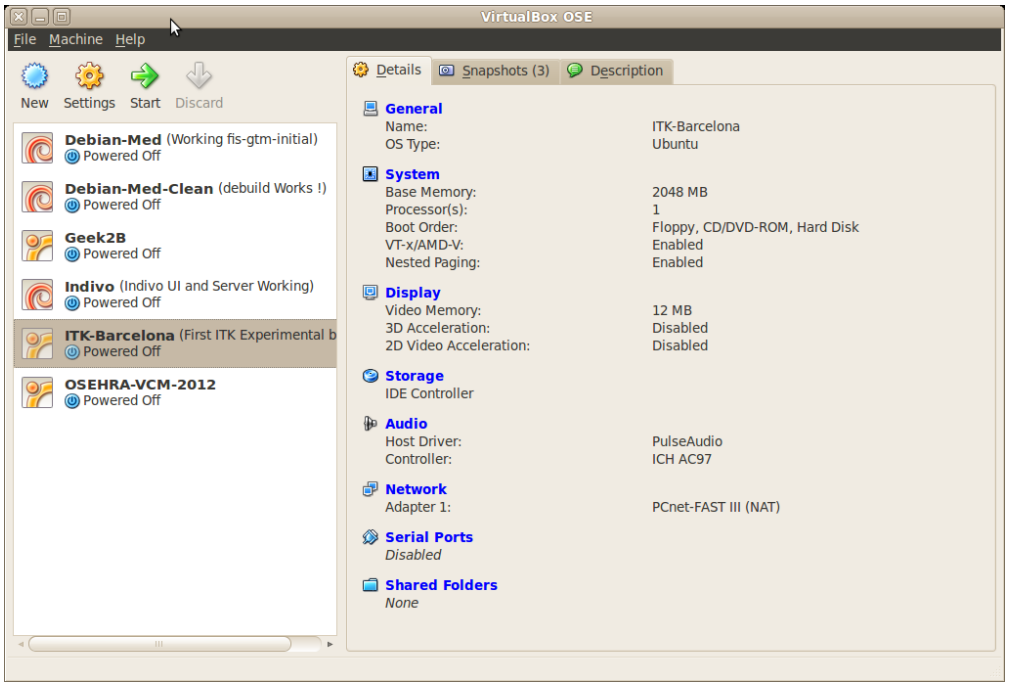
- Select the installer for your platform
- Run it

Alternative Linux Installation

- You can also install VirtualBox by doing:
- sudo apt-get install virtualbox-ose-qt

Importing the Virtual Machine

- Run VirtualBox
- In “File” Menu select “Import Appliance”
- Provide the filename in the DVD / USB stick “VirtualMachine/ITK-Barcelona.ova”
- A progress bar will appear, and when it finishes you should see:



ITKv4

Overview

10 Years Old

$ 5 Million

ARRA Funds

2 Years

Next 10 Years

ITKv4 Team

- GE Global Research

ITKv4 Team

- GE Global Research
- Kitware Inc.

ITKv4 Team

- GE Global Research
- Kitware Inc.
- Harvard University

ITKv4 Team

- GE Global Research
- Kitware Inc.
- Harvard University
- Mayo Clinic

ITKv4 Team

- GE Global Research
- Kitware Inc.
- Harvard University
- Mayo Clinic
- University of Iowa

ITKv4 Team

- GE Global Research
- Kitware Inc.
- Harvard University
- Mayo Clinic
- University of Iowa
- University of Pennsylvania

ITKv4 Team

- GE Global Research
- Kitware Inc.
- Harvard University
- Mayo Clinic
- University of Iowa
- University of Pennsylvania
- CoSMo Software

ITKv4 Team

- GE Global Research
- Kitware Inc.
- Harvard University
- Mayo Clinic
- University of Iowa
- University of Pennsylvania
- CoSMo Software
- National Library of Medicine

A2D2 Team

- Georgetown University

A2D2 Team

- Georgetown University
- University of Utah

A2D2 Team

- Georgetown University
- University of Utah
- UNC - Chape Hill

A2D2 Team

- Georgetown University
- University of Utah
- UNC - Chape Hill
- The Ohio State University

A2D2 Team

- Georgetown University
- University of Utah
- UNC - Chape Hill
- The Ohio State University
- Old Dominion University

A2D2 Team

- Georgetown University
- University of Utah
- UNC - Chape Hill
- The Ohio State University
- Old Dominion University
- Carnegie Mellon University

A2D2 Team

- Georgetown University
- University of Utah
- UNC - Chape Hill
- The Ohio State University
- Old Dominion University
- Carnegie Mellon University
- Harvard University

A2D2 Team

- Georgetown University
- University of Utah
- UNC - Chape Hill
- The Ohio State University
- Old Dominion University
- Carnegie Mellon University
- Harvard University
- University of Utah

A2D2 Team

- Georgetown University
- University of Utah
- UNC - Chape Hill
- The Ohio State University
- Old Dominion University
- Carnegie Mellon University
- Harvard University
- University of Utah
- William and Mary

A2D2 Team

- Georgetown University
- University of Utah
- UNC - Chape Hill
- The Ohio State University
- Old Dominion University
- Carnegie Mellon University
- Harvard University
- University of Utah
- William and Mary
- Kitware Inc.

A2D2 Team

- Georgetown University
- University of Utah
- UNC - Chape Hill
- The Ohio State University
- Old Dominion University
- Carnegie Mellon University
- Harvard University
- University of Utah
- William and Mary
- Kitware Inc.
- Cornell University

Major Changes

Apache 2.0 License

“Patented”
Directory
Removed

New Software Process

Software Process

- Git

Software Process

- Git
- Gerrit

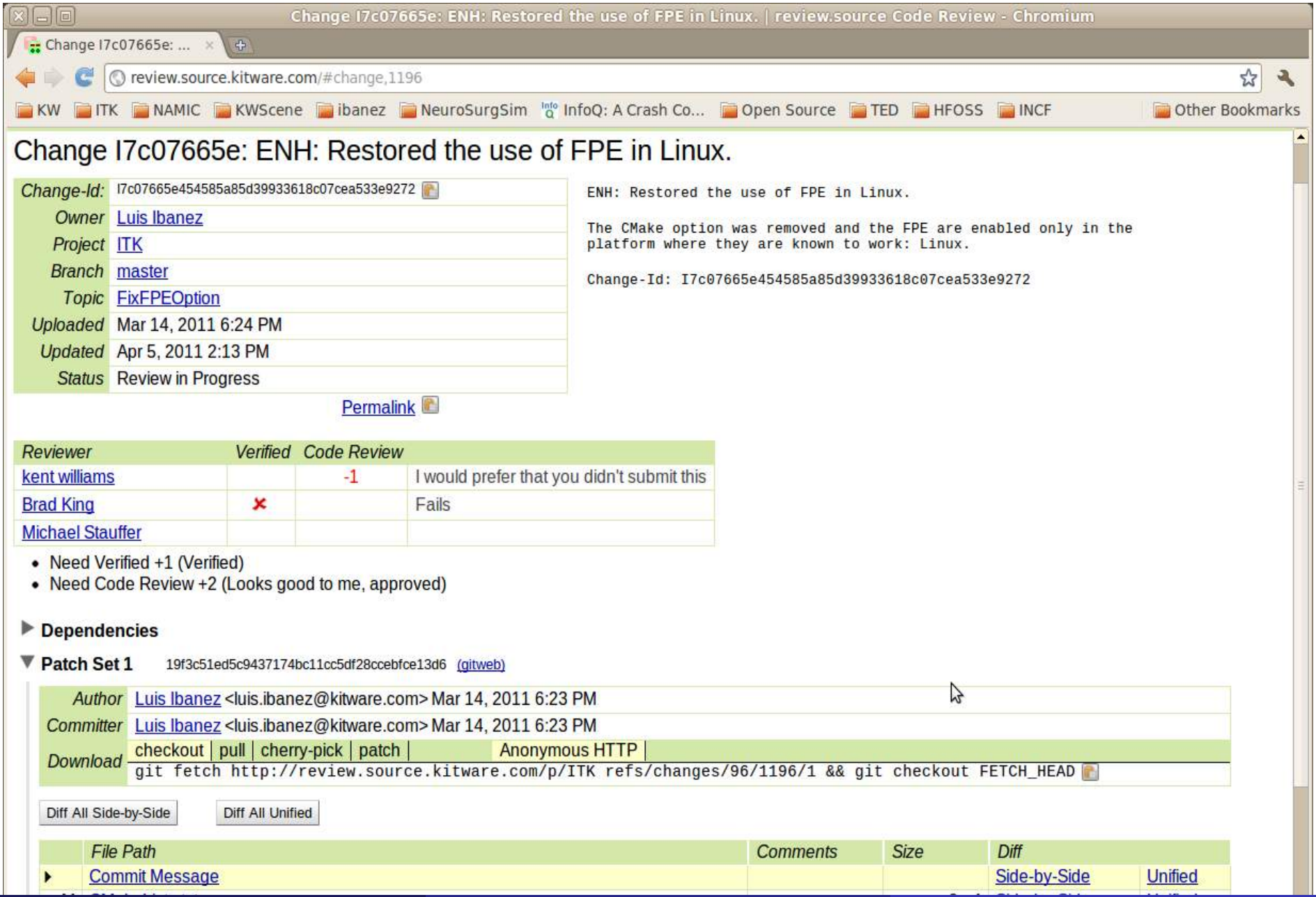
Software Process

- Git
- Gerrit
- cdash@home

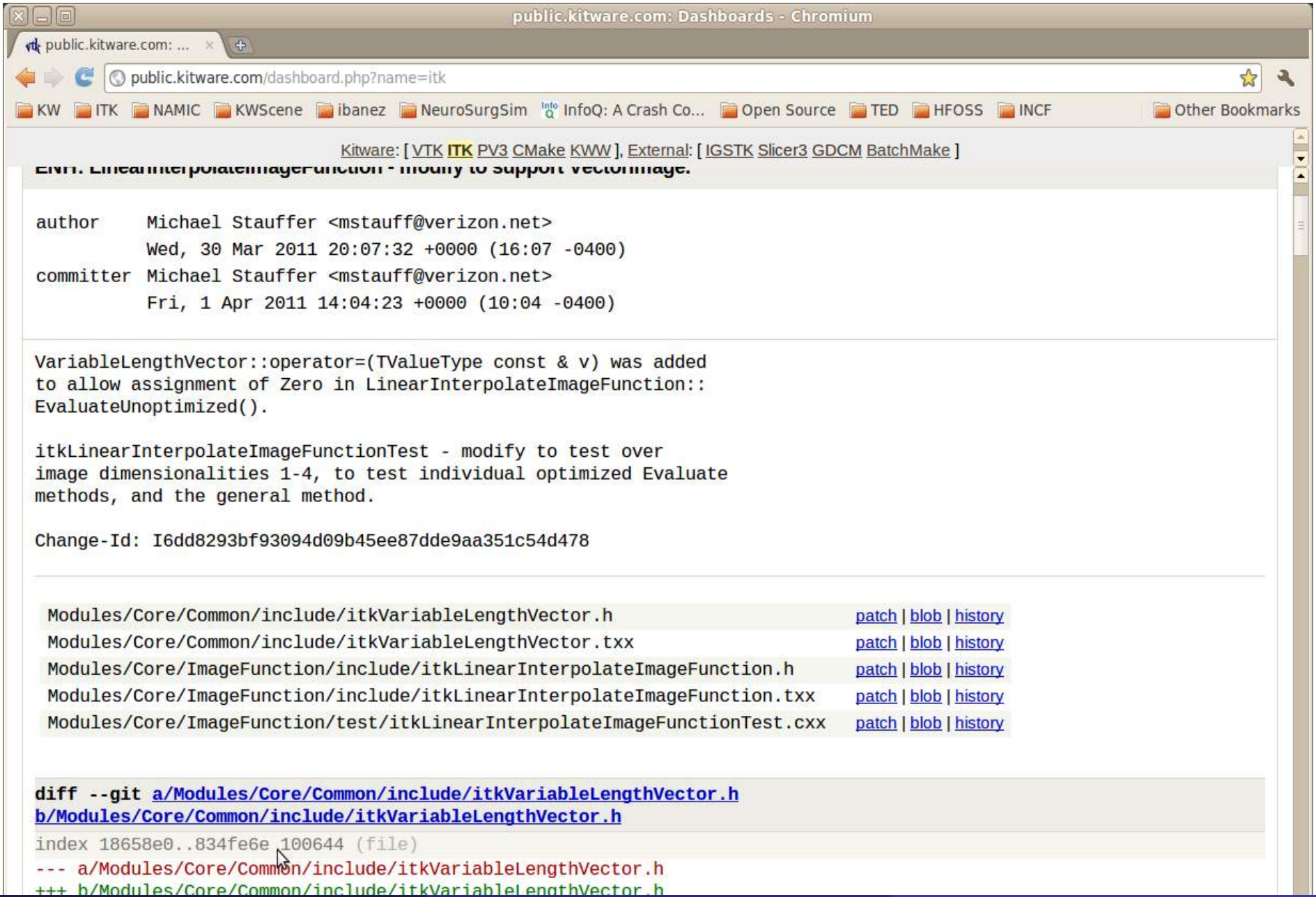
Software Process

- Git
- Gerrit
- cdash@home
- Improved Testing Data

Code Review


Change I7c07665e: ENH: Restored the use of FPE in Linux. | review.source Code Review - Chromium
Change I7c07665e: ...
review.source.kitware.com/#change,1196
KW ITK NAMIC KWScene ibanez NeuroSurgSim InfoQ: A Crash Co... Open Source TED HFOSS INCF Other Bookmarks
Change I7c07665e: ENH: Restored the use of FPE in Linux.
Change-Id: I7c07665e454585a85d39933618c07cea533e9272
Owner: Luis Ibanez
Project: ITK
Branch: master
Topic: FixFPEOption
Uploaded: Mar 14, 2011 6:24 PM
Updated: Apr 5, 2011 2:13 PM
Status: Review in Progress
Permalink
ENH: Restored the use of FPE in Linux.
The CMake option was removed and the FPE are enabled only in the platform where they are known to work: Linux.
Change-Id: I7c07665e454585a85d39933618c07cea533e9272
Reviewer | Verified | Code Review
kent williams | -1 | I would prefer that you didn't submit this
Brad King | ✘ | Fails
Michael Stauffer
Need Verified +1 (Verified)
Need Code Review +2 (Looks good to me, approved)
Dependencies
Patch Set 1 19f3c51ed5c9437174bc11cc5df28ccebfce13d6 (gitweb)
Author: Luis Ibanez <luis.ibanez@kitware.com> Mar 14, 2011 6:23 PM
Committer: Luis Ibanez <luis.ibanez@kitware.com> Mar 14, 2011 6:23 PM
Download: checkout | pull | cherry-pick | patch | Anonymous HTTP
git fetch http://review.source.kitware.com/p/ITK refs/changes/96/1196/1 && git checkout FETCH_HEAD
Diff All Side-by-Side
Diff All Unified
File Path | Comments | Size | Diff
Commit Message
Side-by-Side
Unified

Git Code Repository



Modern C++

Deprecated Compilers

- Visual Studio 6.0, 7.0

Deprecated Compilers

- Visual Studio 6.0, 7.0
- Borland 5.5

Deprecated Compilers

- Visual Studio 6.0, 7.0
- Borland 5.5
- Sun CC < 5.9

Deprecated Compilers

- Visual Studio 6.0, 7.0
- Borland 5.5
- Sun CC < 5.9
- SGI CC

Deprecated Compilers

- Visual Studio 6.0, 7.0
- Borland 5.5
- Sun CC < 5.9
- SGI CC
- MWORKS

Deprecated Compilers

- Visual Studio 6.0, 7.0
- Borland 5.5
- Sun CC < 5.9
- SGI CC
- MWORKS
- Cygwin

Deprecated Compilers

- Visual Studio 6.0, 7.0
- Borland 5.5
- Sun CC < 5.9
- SGI CC
- MWORKS
- Cygwin
- GCC < 3.4

64 Bits

64 Bits

- Since 2008

64 Bits

- Since 2008
- In Windows 64bits:
  unsigned long is 32bits

64 Bits

- Since 2008
- In Windows 64bits:
  unsigned long is 32bits
- Use typedef !

64 Bits

- Since 2008
- In Windows 64bits: unsigned long is 32bits
- Use typedef !
- ITK_USE_64BITS_IDS

ITK Libraries

- ITKCommon, ITKIO, ITKImageIntensity, $\cdots$

ITK Libraries

- ITKCommon, ITKIO, ITKImageIntensity, $\cdots$
- is now:

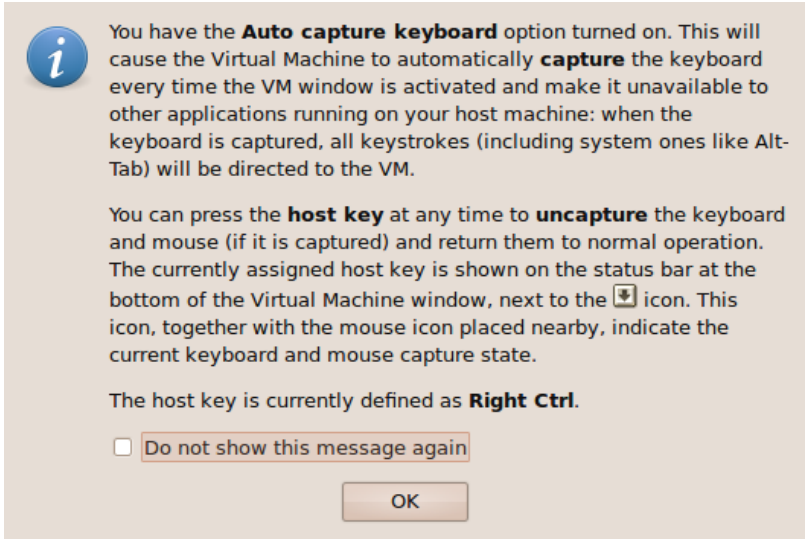
ITK Libraries

- ITKCommon, ITKIO, ITKImageIntensity, $\cdots$
- is now:
- ${ITK_LIBRARIES}

Virtual Machine Preparation II

Booting the Virtual Machine

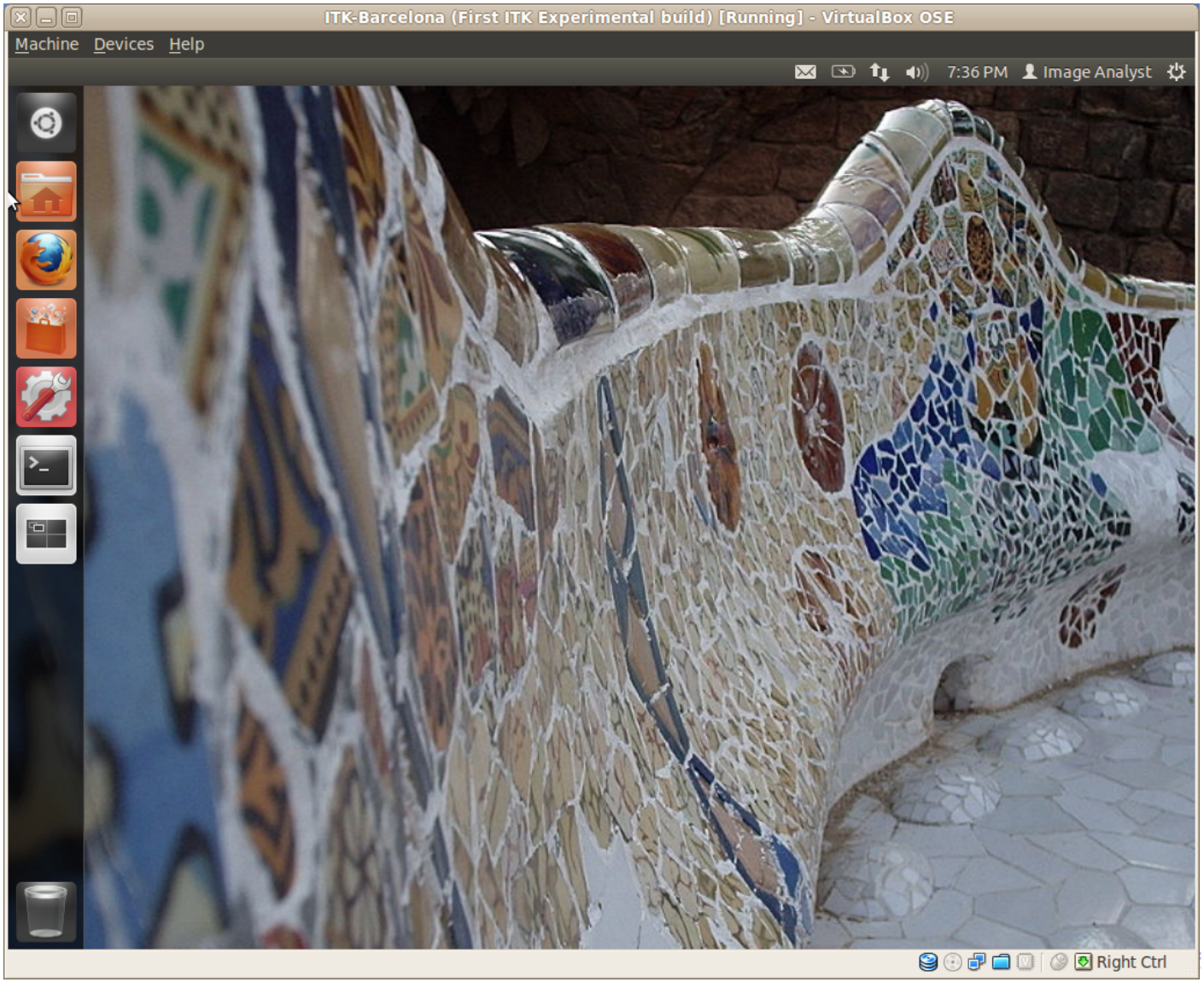
- Click on the “ITK-Barcelona” icon on the left, to select it.
- Click on the Green Arrow at the top “Show”.
- The VM will start to boot and you will see the warning:



- Click “OK”

Booting the Virtual Machine

- The boot sequence should continue and you should see:



Booting the Virtual Machine

- Username: image
- Password: reproducibility

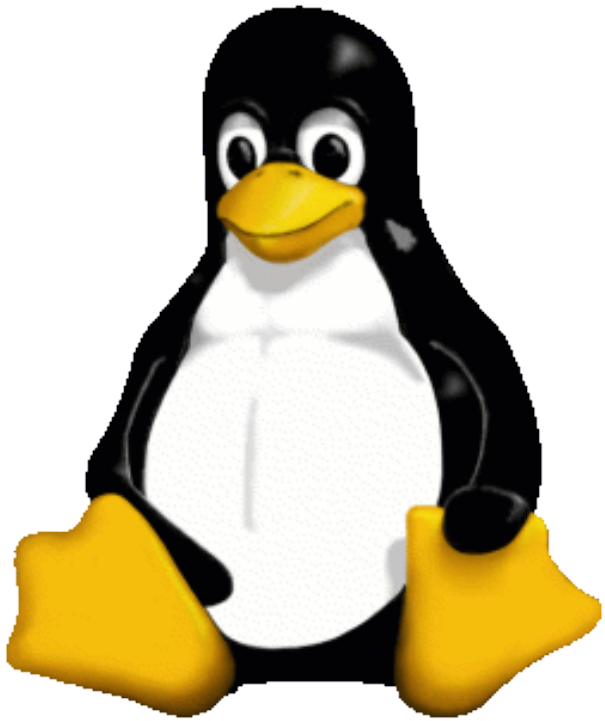
Your Virtual Machine is Ready !

Additional Copies

- The same source trees are available in the DVD / USB key
- Outside of the VirtualBox image

Software Environment

Welcome to Ubuntu GNU/Linux



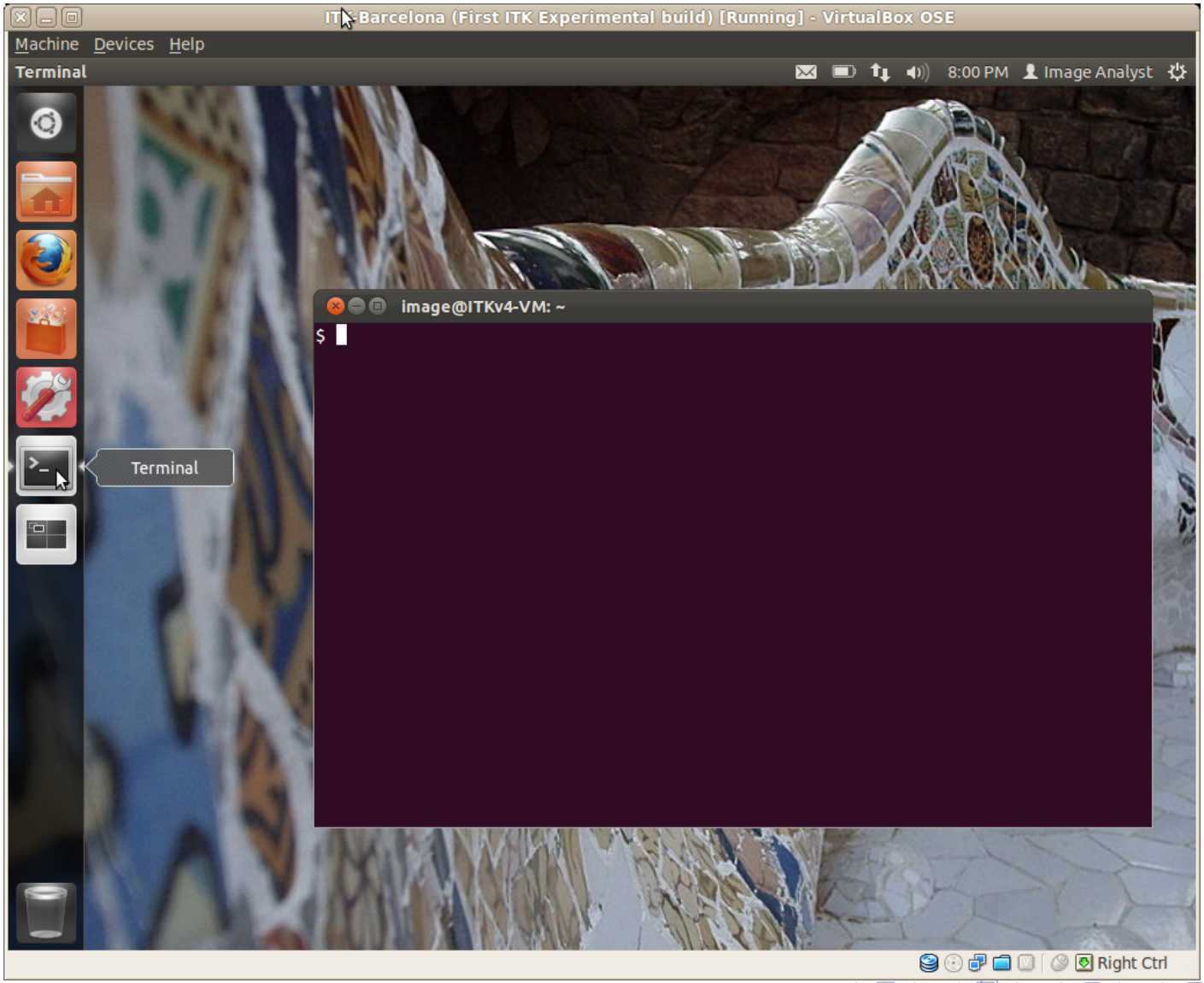
ubuntu

How to take the mouse out of the Virtual Machine

- Hit the **RIGHT CTRL** Key on Windows / Linux Host
- Hit the **LEFT APPLE** Key on Mac Host

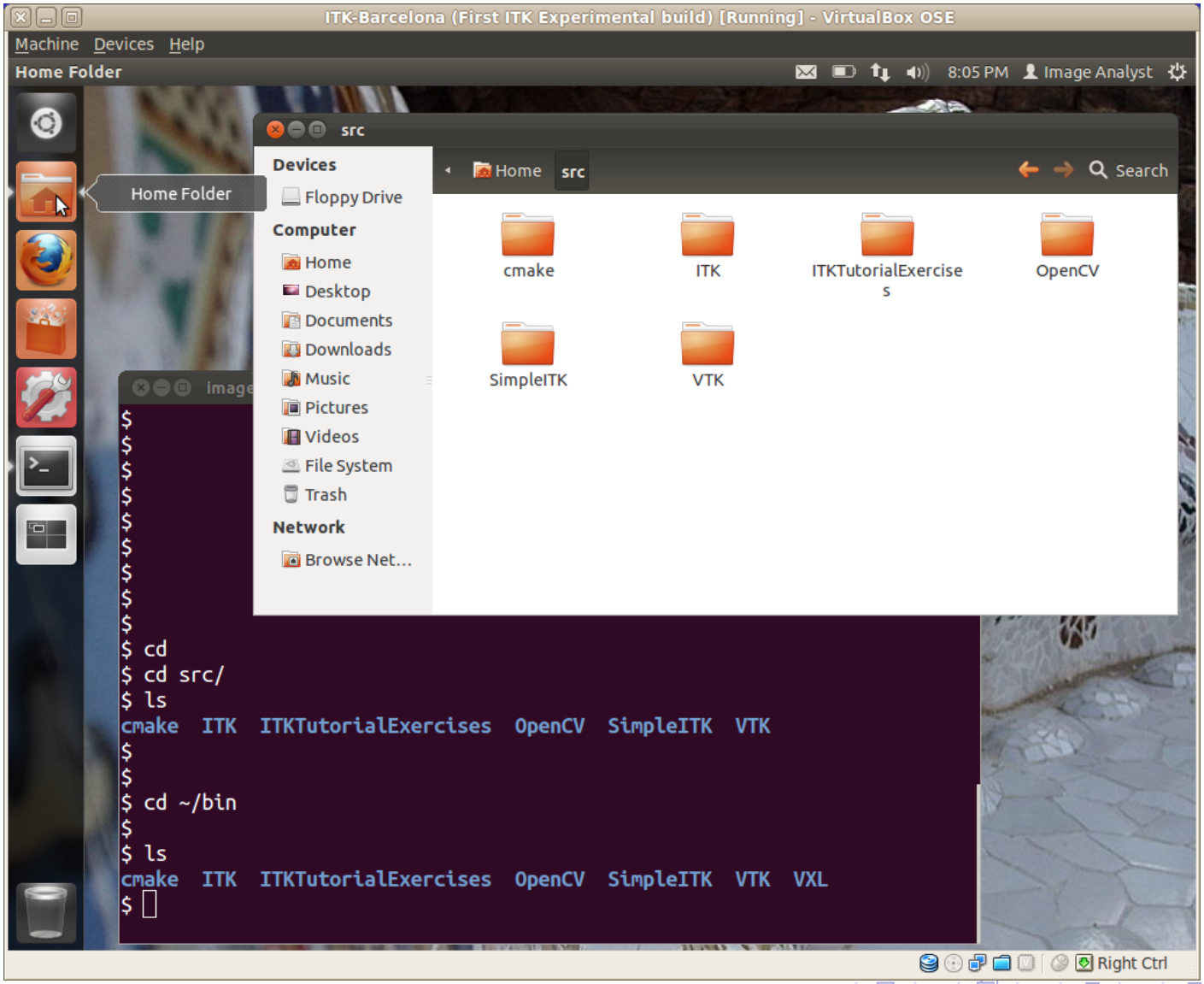
How to Open a Terminal - Icon in Upper Left Corner

To type your command line instructions



How to Navigate Directories

Double Click in Folder Icons in the Desktop



Walk through the directories

- Find source code of exercises

```
cd ~/Documents/ITKv4-TheNextGeneration-Tutorial/Exercises
pwd
ls
nautilus .
```

Walk through the directories

- Find source code of exercises

```
cd ~/Documents/ITKv4-TheNextGeneration-Tutorial/Exercises
pwd
ls
nautilus .
```

- Find binary build of exercises

```
cd ~/bin/Documents/ITKv4-TheNextGeneration-Tutorial/Exerci
pwd
ls
```

The TAB Key is your Friend

- When writing filenames
- Use the TAB key for completions
- No need to type full filenames

How to View Images

- Go to the directory
- Invoke "ImageViewer" application

```
cd ~/data
ImageViewer BrainProtonDensitySlice.png
```

How to View Images

- Go to the directory
- Invoke "ImageViewer" application

```
cd ~/data
ImageViewer BrainProtonDensitySlice.png
```

- Hit "+" key to zoom in

How to View Images

- Go to the directory
- Invoke "ImageViewer" application

```
cd ~/data
ImageViewer BrainProtonDensitySlice.png
```

- Hit "+" key to zoom in
- Hit "-" key to zoom out

How to View Images

- Go to the directory
- Invoke "ImageViewer" application

```
cd ~/data
ImageViewer BrainProtonDensitySlice.png
```

- Hit "+" key to zoom in
- Hit "-" key to zoom out
- Hit ESC key to quit the application

Refactored Registration Framework

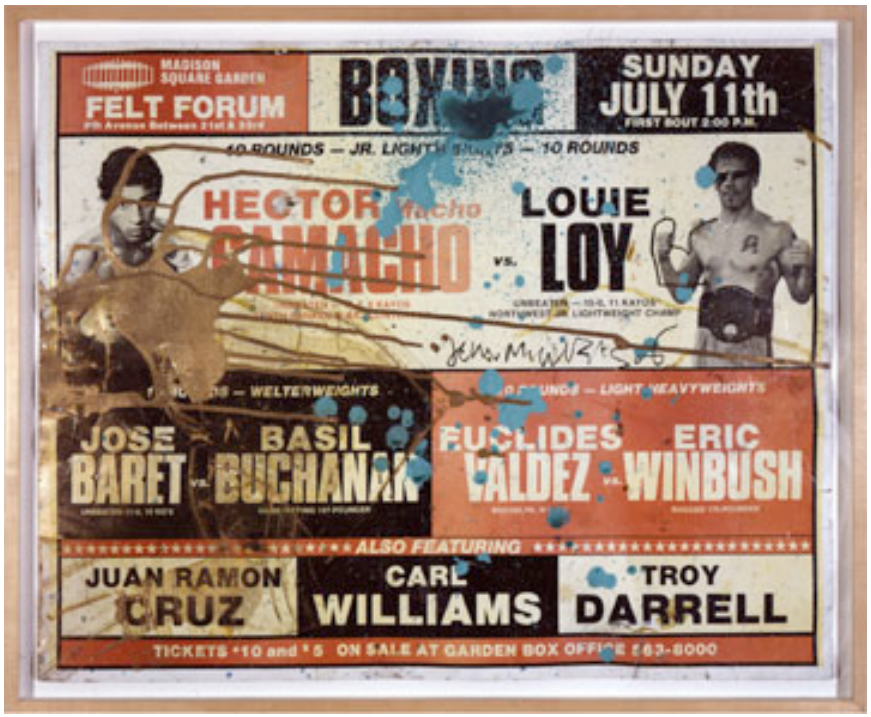
Insight Software Consortium

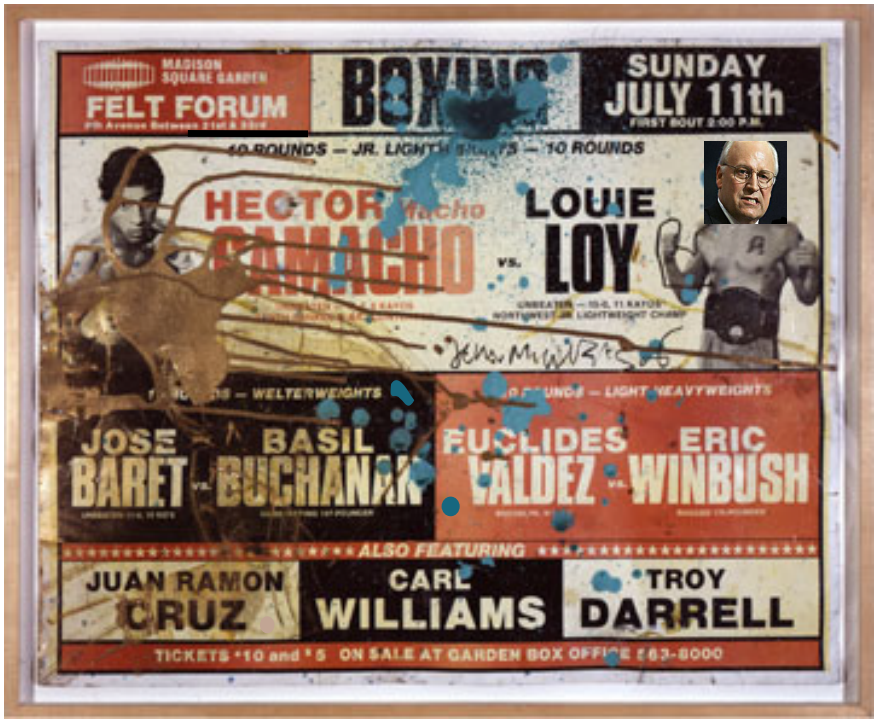
PICSL @ University of Pennsylvania

Brian Avants, Nicholas Tustison, Gang Song,
Baohua Wu, Michael Stauffer, James C. Gee

What is registration?

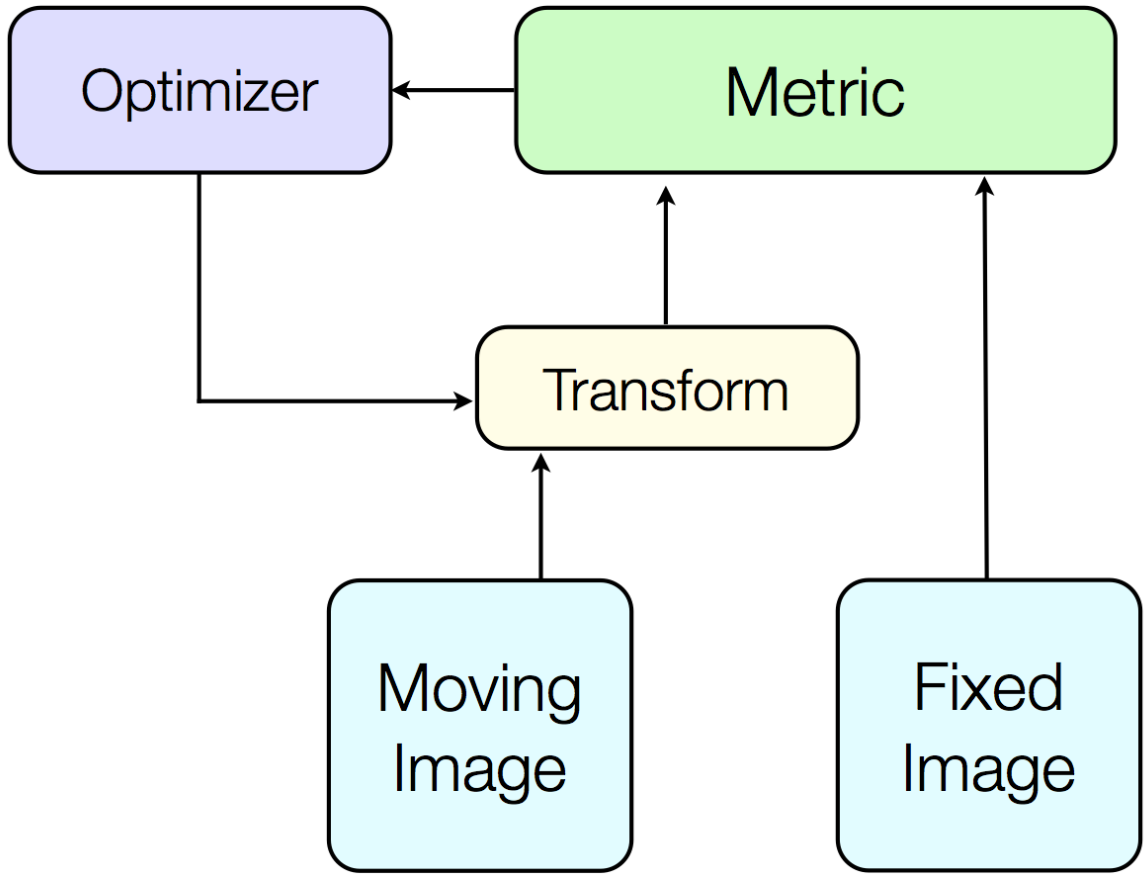
What is registration?



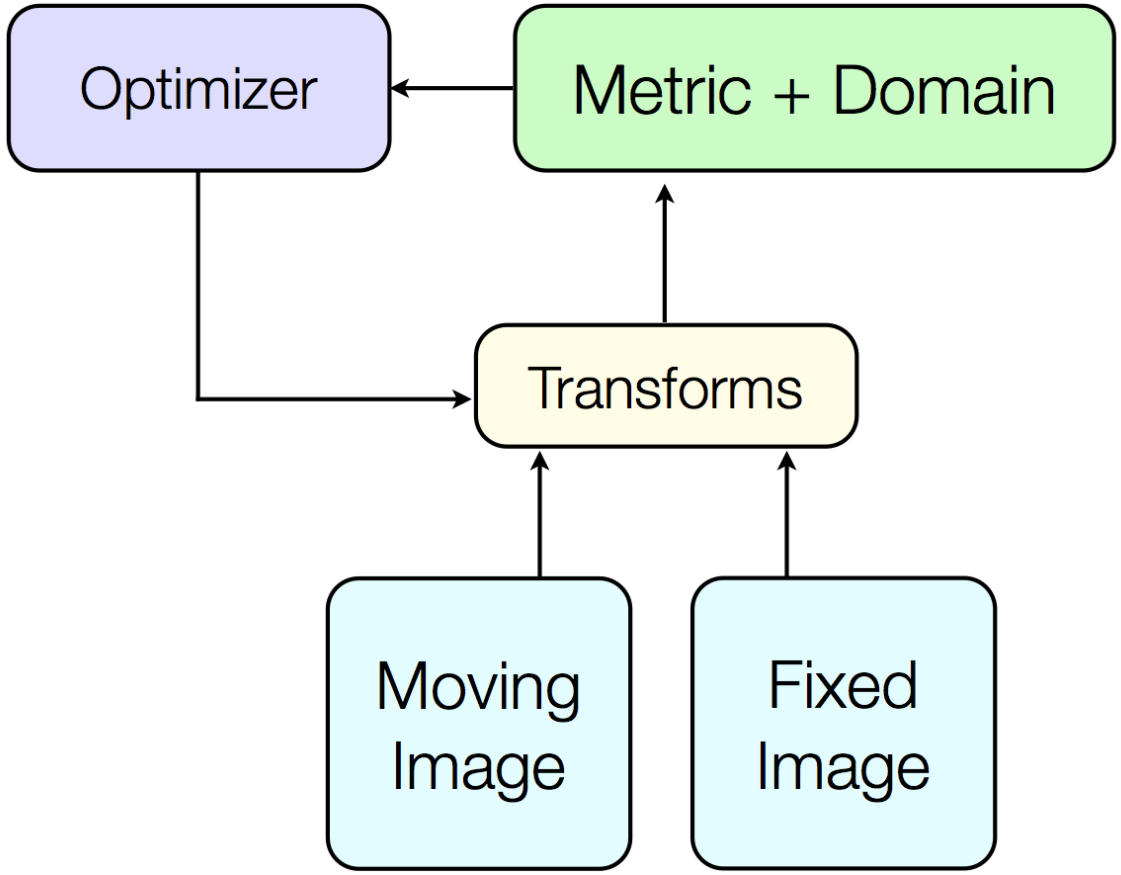


$$\|I(x) - J(\phi(x))\|^2 + R(\phi(x))$$

ITK v3 framework



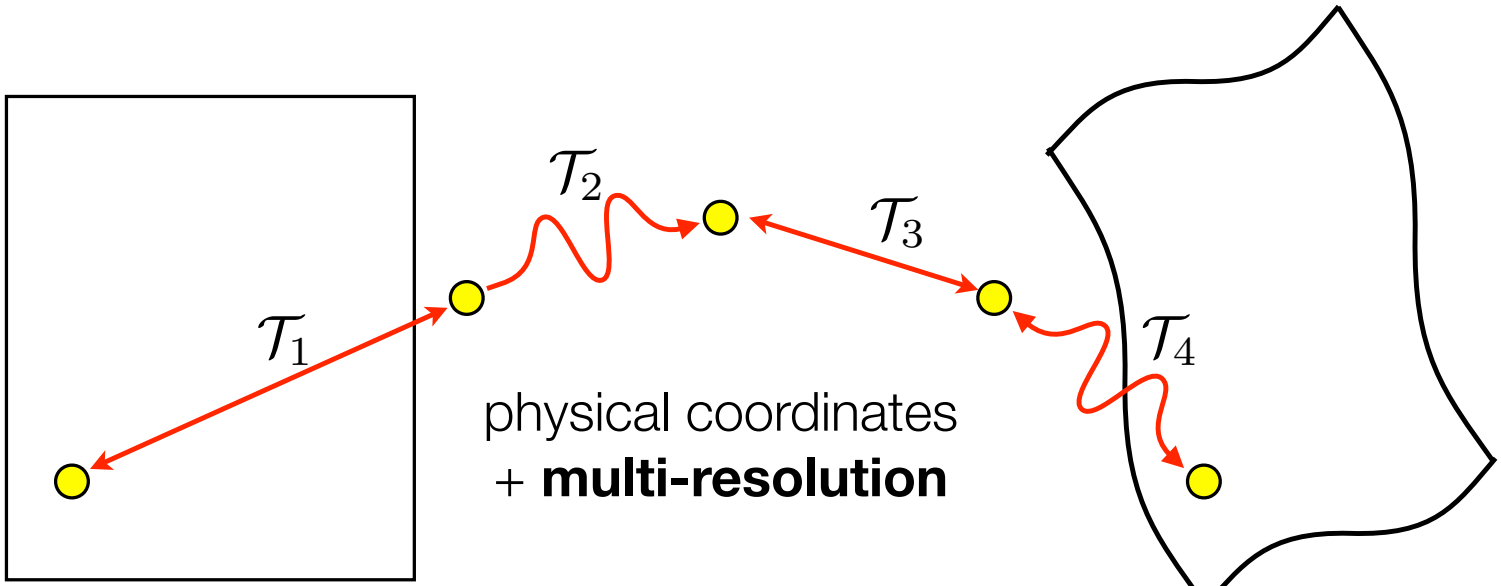
ITK v4 framework



Exercise 1: Look at options for a registration program

```
~/bin/ANTS/bin/antsRegistration --help | less
```

Composite transformations



$$\mathcal{T} = \mathcal{T}_1 \circ \mathcal{T}_2 \circ \mathcal{T}_3 \circ \mathcal{T}_4$$

Transformation Legend

- linear
- deformable
- symmetric deformable

To avoid compounding interpolation error with the concatenation of transformations, never use more than a single interpolation.

Image Registration Refactoring

- Transform operations for vectors & tensors

Image Registration Refactoring

- Transform operations for vectors & tensors
- Unbiased registration

Image Registration Refactoring

- Transform operations for vectors & tensors
- Unbiased registration
- Multi-threading throughout the metrics

Image Registration Refactoring

- Transform operations for vectors & tensors
- Unbiased registration
- Multi-threading throughout the metrics
- Automated parameter scaling

Unified framework

- Local & global transforms treated the same in resampler

Unified framework

- Local & global transforms treated the same in resampler
- Both types available to (revised) optimizers

Unified framework

- Local & global transforms treated the same in resampler
- Both types available to (revised) optimizers
- Revised metrics optimized for these operations

Example Registration

Exercise 2: Perform a simple registration

```
gedit ~/Documents/ITKv4-TheNextGeneration-Tutorial/\
Exercises/Registration/RegistrationExercises.cxx &

cd ~/bin/Documents/ITKv4-TheNextGeneration-Tutorial
make

eog ~/data/BrainProtonDensitySliceBorder20.png &
eog ~/data/BrainProtonDensitySliceR10X13Y17.png &
```

Exercise 2: Perform a simple registration

```
./bin/RegistrationExercises ~/data/BrainProtonDensitySliceBorc
  ~/data/BrainProtonDensitySliceR10X13Y17.png \
  ./Registered.png

eog ./Registered.png
```

Exercise 3: Do not scale the optimizer parameters

- In `RegistrationExercises.cxx`, **line 163**, turn off automatic optimizer paramer scaling.
- What happens and why?

Refactored Level Sets

Insight Software Consortium
Megason Lab, Department of Systems Biology
Harvard Medical School

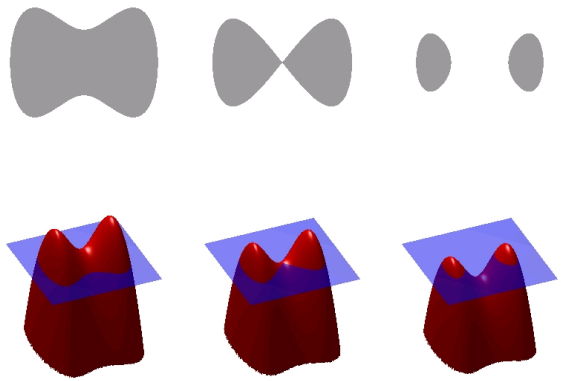
Arnaud Gelas Kishore Mosaliganti Sean Megason

Introduction

What is a level-set function?

Definition

- Implicit function $\phi : \Omega \rightarrow \mathbb{R}$
- If $\phi(p) = 0$, p is on the interface Γ
- If $\phi(p) < 0$, p is inside
- Else p is outside



Level-Set Evolution

- Deforms the level-set function
  - Driven by given PDE

Level-Set Evolution

- Deforms the level-set function
  - Driven by given PDE
    - Regularized Advection

$$\frac{\partial \phi}{\partial \tau} = \alpha \cdot \vec{A}(p) \bullet \vec{\nabla} \phi + \gamma \cdot \text{div} \left(\frac{\vec{\nabla} \phi}{\| \vec{\nabla} \phi \|} \right) \cdot \| \vec{\nabla} \phi \|$$

Advection

$$\alpha \cdot \vec{A}(p) \bullet \vec{\nabla} \phi$$

α: coefficient
$A(p)$: Advection field

Curvature

$$\gamma \cdot \| \vec{\nabla} \phi \| \cdot \text{div} \frac{\vec{\nabla} \phi}{\| \vec{\nabla} \phi \|}$$

γ: coefficient

Level-Set Evolution

- Deforms the level-set function
  - Driven by given PDE
    - Regularized Advection

$$\frac{\partial \phi}{\partial \tau} = \alpha \cdot \vec{A}(p) \bullet \vec{\nabla}\phi + \gamma \cdot \text{div}\left(\frac{\vec{\nabla}\phi}{\|\vec{\nabla}\phi\|}\right) \cdot \|\vec{\nabla}\phi\|$$

    - Regularized Propagation

$$\frac{\partial \phi}{\partial \tau} = \beta \cdot P(p) \cdot \|\vec{\nabla}\phi\| + \gamma \cdot \text{div}\left(\frac{\vec{\nabla}\phi}{\|\vec{\nabla}\phi\|}\right) \cdot \|\vec{\nabla}\phi\|$$

Propagation

$$\beta \cdot P(p) \cdot \|\vec{\nabla}\phi\|$$

β: coefficient
$P(p)$: Propagation field

Curvature

$$\gamma \cdot \|\vec{\nabla}\phi\| \cdot \text{div}\ \frac{\vec{\nabla}\phi}{\|\vec{\nabla}\phi\|}$$

γ: coefficient

Level-Set Evolution

- Deforms the level-set function
  - Driven by given PDE
    - Regularized Advection

$$\frac{\partial \phi}{\partial \tau} = \alpha \cdot \vec{A}(p) \bullet \vec{\nabla}\phi + \gamma \cdot \text{div}\left(\frac{\vec{\nabla}\phi}{\|\vec{\nabla}\phi\|}\right) \cdot \|\vec{\nabla}\phi\|$$

    - Regularized Propagation

$$\frac{\partial \phi}{\partial \tau} = \beta \cdot P(p) \cdot \|\vec{\nabla}\phi\| + \gamma \cdot \text{div}\left(\frac{\vec{\nabla}\phi}{\|\vec{\nabla}\phi\|}\right) \cdot \|\vec{\nabla}\phi\|$$

    - Chan and Vese

$$\frac{\partial \phi}{\partial \tau} = \delta_\epsilon(\phi)\left(-\lambda_{in}\left(I - \mu_{in}\right) + \lambda_{out}\left(I - \mu_{out}\right)\right)$$

Chan And Vese Internal

$$\delta_\epsilon(\phi)\left(\lambda_{in}\left(I - \mu_{in}\right)\right)$$

λ_{in}: coefficient
μ_{in}: Internal Mean

Chan And Vese External

$$\delta_\epsilon(\phi)\left(\lambda_{out}\left(I - \mu_{out}\right)\right)$$

λ_{out}: coefficient
μ_{out}: External Mean

Level-Set Evolution

- Iterative computation
- Topological flexibility

Level Sets: Challenges in Segmentation?

- PDE Term choice
  - Advection terms ?
  - Propagation terms ?
  - Region terms ?
  - Regularization terms ?

Level Sets: Challenges in Segmentation?

- PDE Term choice
  - Advection terms ?
  - Propagation terms ?
  - Region terms ?
  - Regularization terms ?
- Stopping criterion
  - Number of Iterations ?
  - Variation of interface length / area ?
  - Variation of shape area / volume ?

Level Sets: Challenges in Segmentation?

- PDE Term choice
  - Advection terms ?
  - Propagation terms ?
  - Region terms ?
  - Regularization terms ?
- Stopping criterion
  - Number of Iterations ?
  - Variation of interface length / area ?
  - Variation of shape area / volume ?
- PDE Parameters tuning

Level Sets Representation

Discrete

- Dense

Level Sets Representation

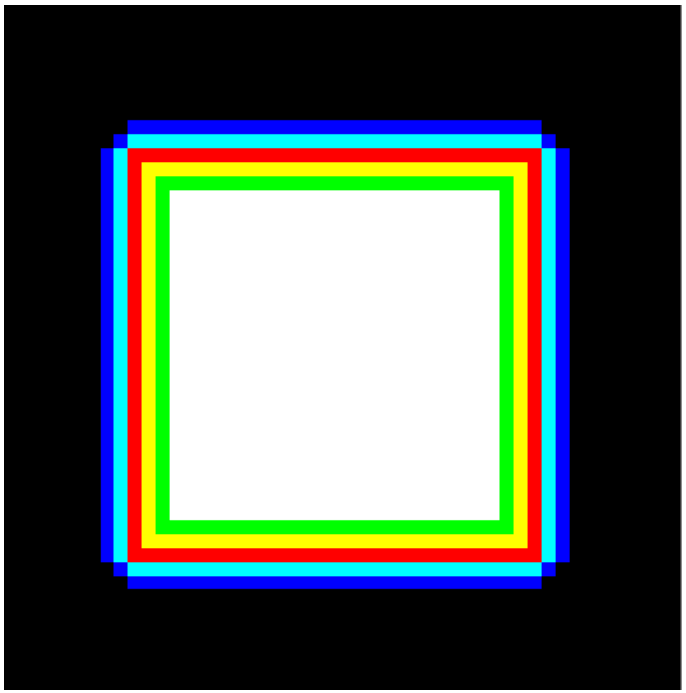
Discrete

- Dense
- Sparse

Level Sets Representation

Discrete

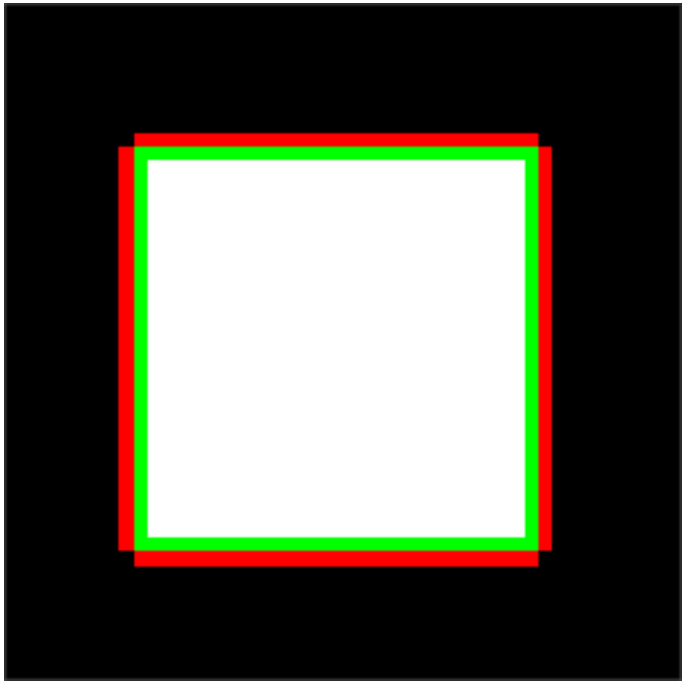
- Dense
- Sparse
  1. Whitaker



Level Sets Representation

Discrete

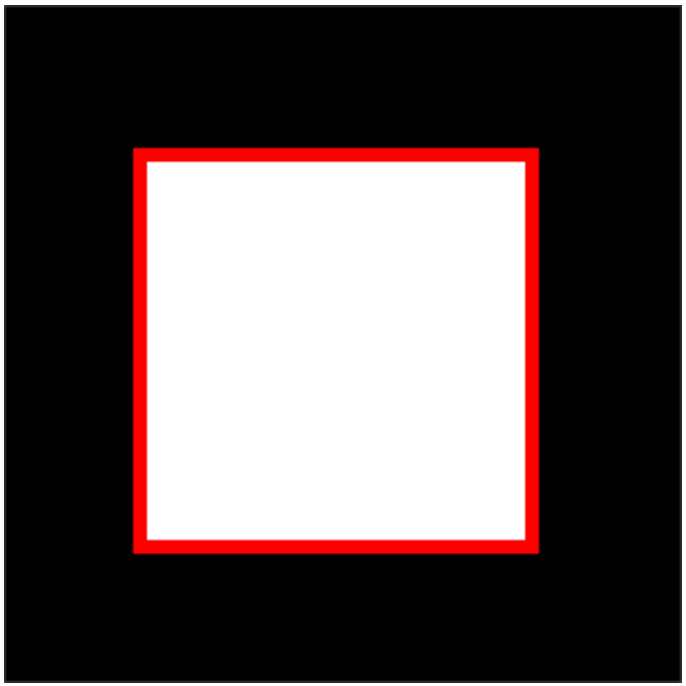
- Dense
- Sparse
  1. Whitaker
  2. Shi



Level Sets Representation

Discrete

- Dense
- Sparse
  1. Whitaker
  2. Shi
  3. Malcolm



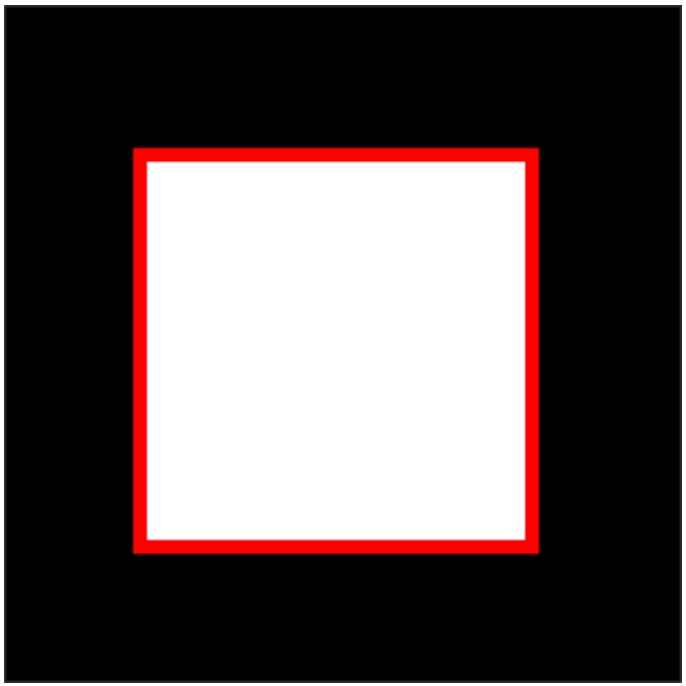
Level Sets Representation

Discrete

- Dense
- Sparse
  1. Whitaker
  2. Shi
  3. Malcolm

Parametric

- Easy integration of new representation



Level Sets Equation

Term

- Contribution for ϕ evolution
- Contribution for time step computation
- Coefficient

Easy to contribute new terms!

Level Sets Equation

Term

- Contribution for ϕ evolution
- Contribution for time step computation
- Coefficient

Easy to contribute new terms!

Level Sets Equation

Term

- Contribution for ϕ evolution
- Contribution for time step computation
- Coefficient

Easy to contribute new terms!

TermContainer

- Represent a given PDEs
- Mix of any term
- Independent of the representation

Easy to contribute new PDEs!

Level Sets Equation

Term

- Contribution for ϕ evolution
- Contribution for time step computation
- Coefficient

Easy to contribute new terms!

TermContainer

- Represent a given PDEs
- Mix of any term
- Independent of the representation

Easy to contribute new PDEs!

Other Features

- N Level-Sets function evolving at the same time
- Geometrical Constraints

How to run the example?

```
cd ~/bin/Documents/ITKv4-TheNextGeneration-Tutorial/bin/

./SingleLevelSetWhitaker
    <InputImage>
    <NumberOfIterations>
    <Visualization (0 or 1)>
    <OutputImage>

./SingleLevelSetWhitaker
    ~/data/cells.png
    800
    1
    cells_segmented.mha
```

Example
Code Walk-Through

Create a level-set function from binary mask

```
53  const unsigned int Dimension = 2;



96  typedef float PixelType;

98  typedef itk::WhitakerSparseLevelSetImage<
99    PixelType, Dimension > WhitakerSparseLevelSetImageType;

101 typedef itk::BinaryImageToLevelSetImageAdaptor<
102  InputImageType, WhitakerSparseLevelSetImageType > BinaryToSparseAdaptorType;

104 BinaryToSparseAdaptorType::Pointer adaptor = BinaryToSparseAdaptorType::New();
105 adaptor->SetInputImage( binaryImage );
106 adaptor->Initialize();



110 typedef BinaryToSparseAdaptorType::LevelSetType SparseLevelSetType;

112 SparseLevelSetType::Pointer levelSet = adaptor->GetLevelSet();
```

Create a domain for the level-set function

```
118  typedef itk::IdentifierType          IdentifierType;
119  typedef std::list< IdentifierType > IdListType;

121  IdListType listIds;
122  listIds.push_back( 1 );



126  typedef itk::Image< IdListType, Dimension >                IdListImageType;
127  IdListImageType::Pointer idImage = IdListImageType::New();
128  idImage->SetRegions( inputImage->GetLargestPossibleRegion() );
129  idImage->Allocate();
130  idImage->FillBuffer( listIds );

132  typedef itk::Image< short, Dimension >                      CacheImageType;
133  typedef itk::LevelSetDomainMapImageFilter< IdListImageType, CacheImageType >
134                                                             DomainMapImageFilterType;
135  DomainMapImageFilterType::Pointer domainMapFilter = DomainMapImageFilterType::New();
136  domainMapFilter->SetInput( idImage );
137  domainMapFilter->Update();
```

Setting up the level-set container

```
141  typedef SparseLevelSetType::OutputRealType LevelSetOutputRealType;

143  typedef itk::SinRegularizedHeavisideStepFunction<
144    LevelSetOutputRealType, LevelSetOutputRealType > HeavisideFunctionBaseType;
145  HeavisideFunctionBaseType::Pointer heaviside = HeavisideFunctionBaseType::New();
146  heaviside->SetEpsilon( 1.0 );



149  typedef itk::LevelSetContainer<
150    IdentifierType, SparseLevelSetType > LevelSetContainerType;

152  LevelSetContainerType::Pointer lscontainer = LevelSetContainerType::New();
153  lscontainer->SetHeaviside( heaviside );
154  lscontainer->SetDomainMapFilter( domainMapFilter );

156  lscontainer->AddLevelSet( 0, levelSet );
```

Creating PDE Terms

- Chan and Vese internal term

```
163  typedef itk::LevelSetEquationChanAndVeseInternalTerm<
164    InputImageType, LevelSetContainerType > InternalTermType;

166  InternalTermType::Pointer cvInternalTerm0 = InternalTermType::New();
167  cvInternalTerm0->SetInput( inputImage );
168  cvInternalTerm0->SetCoefficient( 1.0 );
169  cvInternalTerm0->SetCurrentLevelSetId( 0 );
170  cvInternalTerm0->SetLevelSetContainer( lscontainer );
```

- Chan and Vese external term

```
174  typedef itk::LevelSetEquationChanAndVeseExternalTerm<
175    InputImageType, LevelSetContainerType > ExternalTermType;

177  ExternalTermType::Pointer cvExternalTerm0 = ExternalTermType::New();
178  cvExternalTerm0->SetInput( inputImage );
179  cvExternalTerm0->SetCoefficient( 1.0 );
180  cvExternalTerm0->SetCurrentLevelSetId( 0 );
181  cvExternalTerm0->SetLevelSetContainer( lscontainer );
```

Setting up PDE

```
187  typedef itk::LevelSetEquationTermContainerBase<
188    InputImageType, LevelSetContainerType > TermContainerType;

190  TermContainerType::Pointer termContainer0 = TermContainerType::New();
191  termContainer0->SetInput( inputImage );
192  termContainer0->SetLevelSetContainer( lscontainer );

194  termContainer0->AddTerm( 0, cvInternalTerm0 );
195  termContainer0->AddTerm( 1, cvExternalTerm0 );



199  typedef itk::LevelSetEquationContainerBase< TermContainerType > EquationContainerType;
200  EquationContainerType::Pointer equationContainer = EquationContainerType::New();
201  equationContainer->AddEquation( 0, termContainer0 );
202  equationContainer->SetLevelSetContainer( lscontainer );
```

Stopping criterion

```
204  typedef itk::LevelSetEvolutionNumberOfIterationsStoppingCriterion<
205    LevelSetContainerType > StoppingCriterionType;

207  StoppingCriterionType::Pointer criterion = StoppingCriterionType::New();
208  criterion->SetNumberOfIterations( atoi( argv[2]) );
```

Starts the evolution

- Set a stopping criterion
- Evolve

```
226  IterationUpdateCommandType::Pointer iterationUpdateCommand = IterationUpdateCommand
227  iterationUpdateCommand->SetFilterToUpdate( visualizer );
228  iterationUpdateCommand->SetUpdatePeriod( 1 );

230  if( atoi( argv[3] ) == 1 )
231    {
232    evolution->AddObserver( itk::IterationEvent(), iterationUpdateCommand );
233    }

235  evolution->SetStoppingCriterion( criterion );
```

Exercise: Add a curvature term to LevelSetExercise1.cxx

Wrap-up

- Send us feedback (good or bad)
  - arnaudgelas@gmail.com
  - kishore_mosaliganti@hms.harvard.edu
  - sean_megason@hms.harvard.edu
  - insight-users@public.kitware.com

ITK OpenCV Bridge

OpenCV Overview

- What is OpenCV? OpenCV is . . .
  - an open source computer vision library.
  - written in C, but has C++ and Python APIs.
  - released under a BSD license.
  - supported and guided by Willow Garage.
  - found at `http://opencv.willowgarage.com/wiki/`.

OpenCV Overview

- What is OpenCV? OpenCV is . . .
  - an open source computer vision library.
  - written in C, but has C++ and Python APIs.
  - released under a BSD license.
  - supported and guided by Willow Garage.
  - found at `http://opencv.willowgarage.com/wiki/`.
- We will be using OpenCV 2.4 (from subversion).

Introduction

- ITK Module for working with other libraries
- Moving frame and/or video data between OpenCV and ITK
- Bring biomedical and computer vision folks together

Design Choices

- OpenCV users and ITK users should both be comfortable
- Image to image utility functions
- cv::Source to itk::VideoStream
- Video Group Modules: BridgeOpenCV,BridgeVXL,Core,Filterig,IO

Basic Image Filtering itk::image & cv:mat

Include Header Files

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- We include ITK and OpenCV headers (like before):

```
20 #include <opencv2/imgproc/imgproc.hpp>
21 #include <opencv2/highgui/highgui.hpp>


23 #include <itkImage.h>
24 #include <itkMedianImageFilter.h>
```

- We also need to include the bridge header:

```
25 #include <itkOpenCVImageBridge.h>
```

Basic Layout

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- The basic layout of this file is the same as the OpenCV Examples:

```
27 int main ( int argc, char **argv )
28 {
29   if( argc < 2 )
30   {
31     std::cout << "Usage: "<< argv[0] <<" input_image output_image"<<std::endl;
32     return -1;
33   }

35   cv::Mat inputImage = cv::imread( argv[1] );


69   if(argc < 3)
70   {
71     std::string windowName = "Exercise 1: Basic Filtering in OpenCV & ITK";
72     cv::namedWindow( windowName, CV_WINDOW_FREERATIO);
73     cvResizeWindow( windowName.c_str(), resultImage.cols, resultImage.rows+50 );
74     cv::Mat scaled;
75     resultImage.convertTo( scaled, CV_8UC1 );
76     cv::imshow( windowName, scaled );
77     cv::waitKey();
78   }
79   else
80   {
81     cv::imwrite( argv[2], resultImage );
82   }

84   return 0;
```

Adding ITK

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- The type definitions should also be familiar from the ITK Material:

```
38  typedef unsigned char                                 OutputPixelType;
39  const unsigned int Dimension =                        2;
40  typedef itk::Image< InputPixelType, Dimension >       InputImageType;
41  typedef itk::Image< OutputPixelType, Dimension >      OutputImageType;
42  typedef itk::OpenCVImageBridge                        BridgeType;
43  typedef itk::MedianImageFilter< InputImageType, OutputImageType >
44                                                        FilterType;
```

- However, notice the bridge class. It contains the conversion function between OpenCV and ITK.

```
42  typedef itk::OpenCVImageBridge                        BridgeType;
```

From OpenCV to ITK

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- We call our conversion function to go from a cv::Mat to an itk::Image

```
47    BridgeType::CVMatToITKImage< InputImageType >( inputImage );
```

Filtering with ITK

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- The median filtering is normal ITK code, but we do not connect our output to a writer

```
49  FilterType::Pointer filter = FilterType::New();
50  InputImageType::SizeType neighborhoodRadius;
51  neighborhoodRadius[0] = 9;
52  neighborhoodRadius[1] = 9;
53  filter->SetRadius( neighborhoodRadius );

55  filter->SetInput( itkImage );
56  try
57    {
58    filter->Update();
59    }
60  catch( itk::ExceptionObject & excp )
61    {
62    std::cerr << excp << std::endl;
63    return EXIT_FAILURE;
64    }
```

Filtering with ITK

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- The median filtering is normal ITK code, but we do not connect our output to a writer

```
49  FilterType::Pointer filter = FilterType::New();
50  InputImageType::SizeType neighborhoodRadius;
51  neighborhoodRadius[0] = 9;
52  neighborhoodRadius[1] = 9;
53  filter->SetRadius( neighborhoodRadius );

55  filter->SetInput( itkImage );
56  try
57    {
58    filter->Update();
59    }
60  catch( itk::ExceptionObject & excp )
61    {
62    std::cerr << excp << std::endl;
63    return EXIT_FAILURE;
64    }
```

- Instead, we set it to our conversion function

```
66  cv::Mat resultImage =
67    BridgeType::ITKImageToCVMat< OutputImageType >( filter->GetOutput() );
```

Running the Example

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- Run the example with the following command

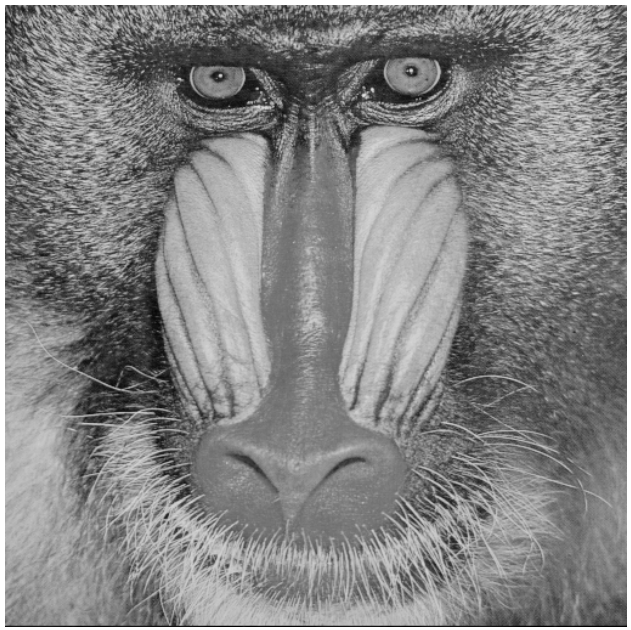
```
cd ~/bin/Documents/ITKv4-TheNextGeneration-Tutorial/
./BasicFilteringITKOpenCVBridge     \
~/data/mandrillgray.png             \
./mandrillgrayMedian.png

 eog .mandrillgrayMedian.png
```

Viewing the Results

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- Running the example the same way as before, we see a nicely median-filtered image.



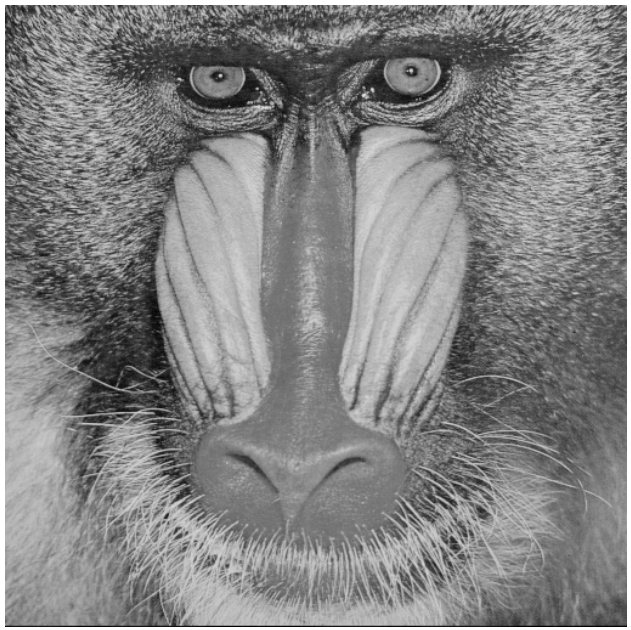
Original

Median Filter

Viewing the Results

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- Running the example the same way as before, we see a nicely median-filtered image.



Original

Median Filter

- Now, the fun part. Let's modify our example to use Curvature Flow, an anisotropic diffusion filter built into ITK.

Exercise 1

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- Hint 1: Curvature Flow requires `float` as the output pixel type.

Exercise 1

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridge.cxx

- Hint 1: Curvature Flow requires `float` as the output pixel type.
- Hint 2: Curvature Flow does not take a radius parameter. It's salient functions are:

```
50   filter->SetTimeStep( 0.5 );
51   filter->SetNumberOfIterations( 20 );
```

Exercise 1: Answer

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridgeAnswer.cxx

- First, change the included filter

```
24 #include <itkCurvatureFlowImageFilter.h>
```

Exercise 1: Answer

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridgeAnswer.cxx

- First, change the included filter

```
24 #include <itkCurvatureFlowImageFilter.h>
```

- The type definitions also need to change to reflect the new filter and output image types.

```
37  typedef unsigned char                                 InputPixelType;
38  typedef float                                         OutputPixelType;
39  const unsigned int Dimension =                        2;
40  typedef itk::Image< InputPixelType, Dimension >       InputImageType;
41  typedef itk::Image< OutputPixelType, Dimension >      OutputImageType;
42  typedef itk::OpenCVImageBridge                        BridgeType;
43  typedef itk::CurvatureFlowImageFilter< InputImageType, OutputImageType >
44                                                        FilterType;
```

Exercise 1: Answer

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridgeAnswer.cxx

- The semantics of calling the filter also has to change:

```
50   filter->SetTimeStep( 0.5 );
51   filter->SetNumberOfIterations( 20 );
```

Exercise 1: Answer

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridgeAnswer.cxx

- The semantics of calling the filter also has to change:

```
50   filter->SetTimeStep( 0.5 );
51   filter->SetNumberOfIterations( 20 );
```

- That's it! Now you're using a "better" blurring scheme.

Exercise 1: Answer

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridgeAnswer.cxx

- The final pipeline should look like this:

```
35  cv::Mat inputImage = cv::imread( argv[1] );


46  InputImageType::Pointer itkImage =
47    BridgeType::CVMatToITKImage< InputImageType >( inputImage );

49  FilterType::Pointer filter = FilterType::New();
50  filter->SetTimeStep( 0.5 );
51  filter->SetNumberOfIterations( 20 );

53  filter->SetInput( itkImage );
54  try
55    {
56    filter->Update();
57    }
58  catch( itk::ExceptionObject & excp )
59    {
60    std::cerr << excp << std::endl;
61    return EXIT_FAILURE;
62    }
```

Running the Answer

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridgeAnswer.cxx

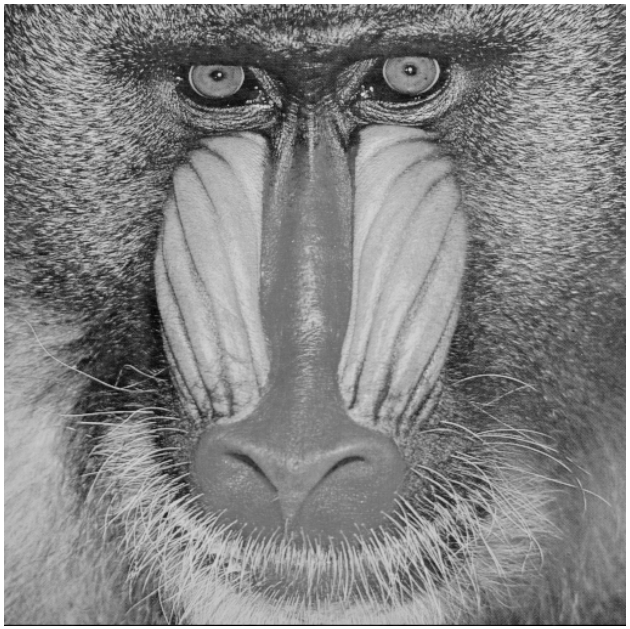
- Run the answer with the following command

```
./BasicFilteringITKOpenCVBridgeAnswer      \
~/data/mandrillgray.png                    \
./mandrillgrayCurvatureFlow.png
```

Exercise 1: Answer

ITKOpenCVBridge/exercise1/BasicFilteringITKOpenCVBridgeAnswer.cxx

- The results should look like this:



Original

Median Filter

Curvature Flow

Basic Video Filtering

- Video filtering is similar to image filtering, except
  - use a `VideoCapture` object to read a video file.
  - loop over each frame in the video and process each one.
  - display or encode each output frame within the loop.

Video Filters

- Support video processing natively in ITK
- Standard framework for multi-frame filters
- Use ITK's library of image filters in video context

More Tutorial Material

- More excercises can be found at: https://github.com/InsightSoftwareConsortium/ITK-OpenCV-Bridge-Tutorial

SimpleITK

- Hiding Templates

- Hiding Templates
- Smarter IO

SimpleITK

- Hiding Templates
- Smarter IO
- Procedural Programming

SimpleITK

- Hiding Templates
- Smarter IO
- Procedural Programming
- Easier Wrapping

SimpleITK

- Hiding Templates
- Smarter IO
- Procedural Programming
- Easier Wrapping
- Binary Distribution

SimpleITK

- Hiding Templates
- Smarter IO
- Procedural Programming
- Easier Wrapping
- Binary Distribution
- See SimpleITK Tutorial. . .

Modularization

Why Modularization

- Growth management

Why Modularization

- Growth management
- Software quality improvement

Why Modularization

- Growth management
- Software quality improvement
- Facilitating add-ons

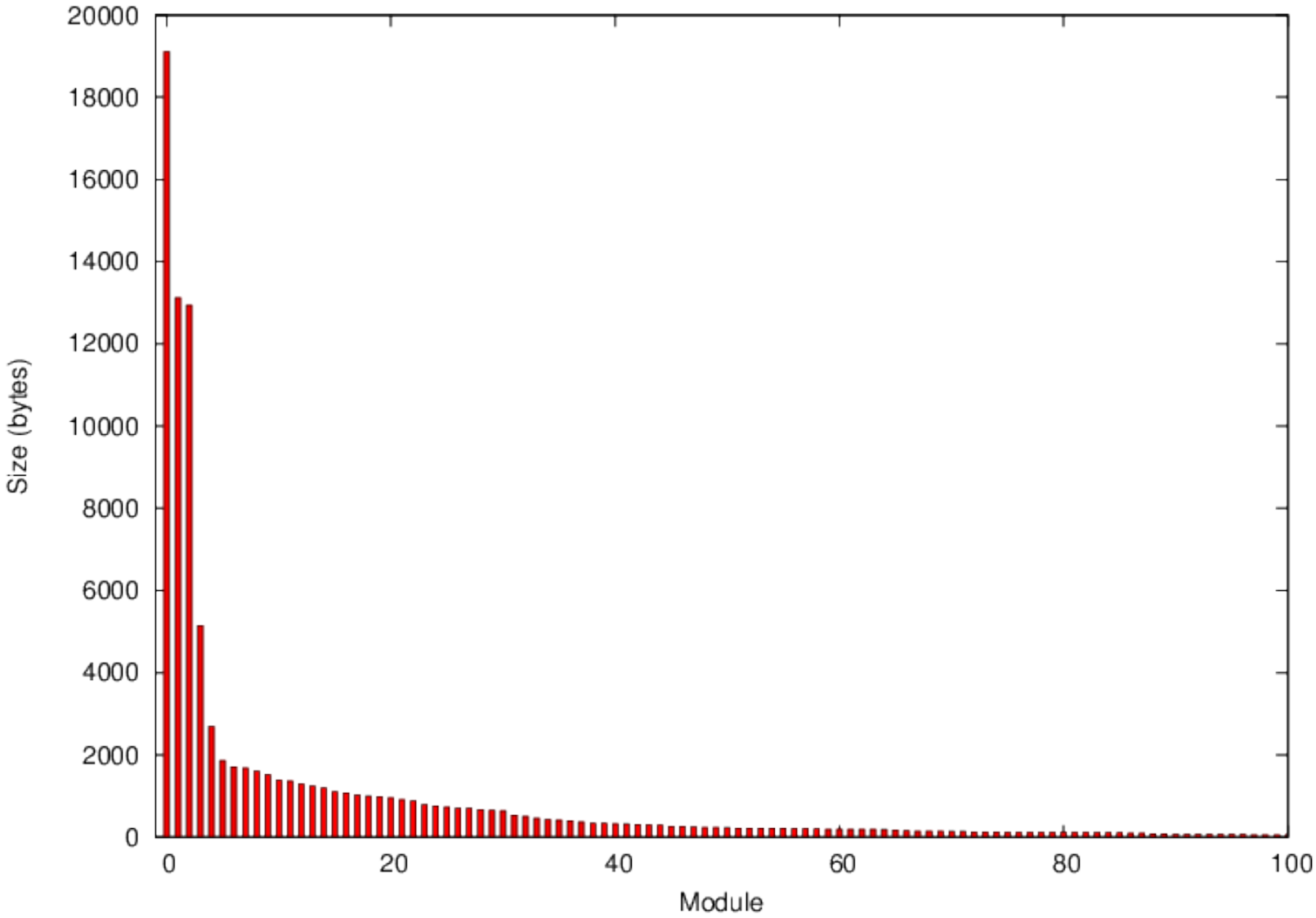
Why Modularization

- Growth management
- Software quality improvement
- Facilitating add-ons
- Separate third party libraries

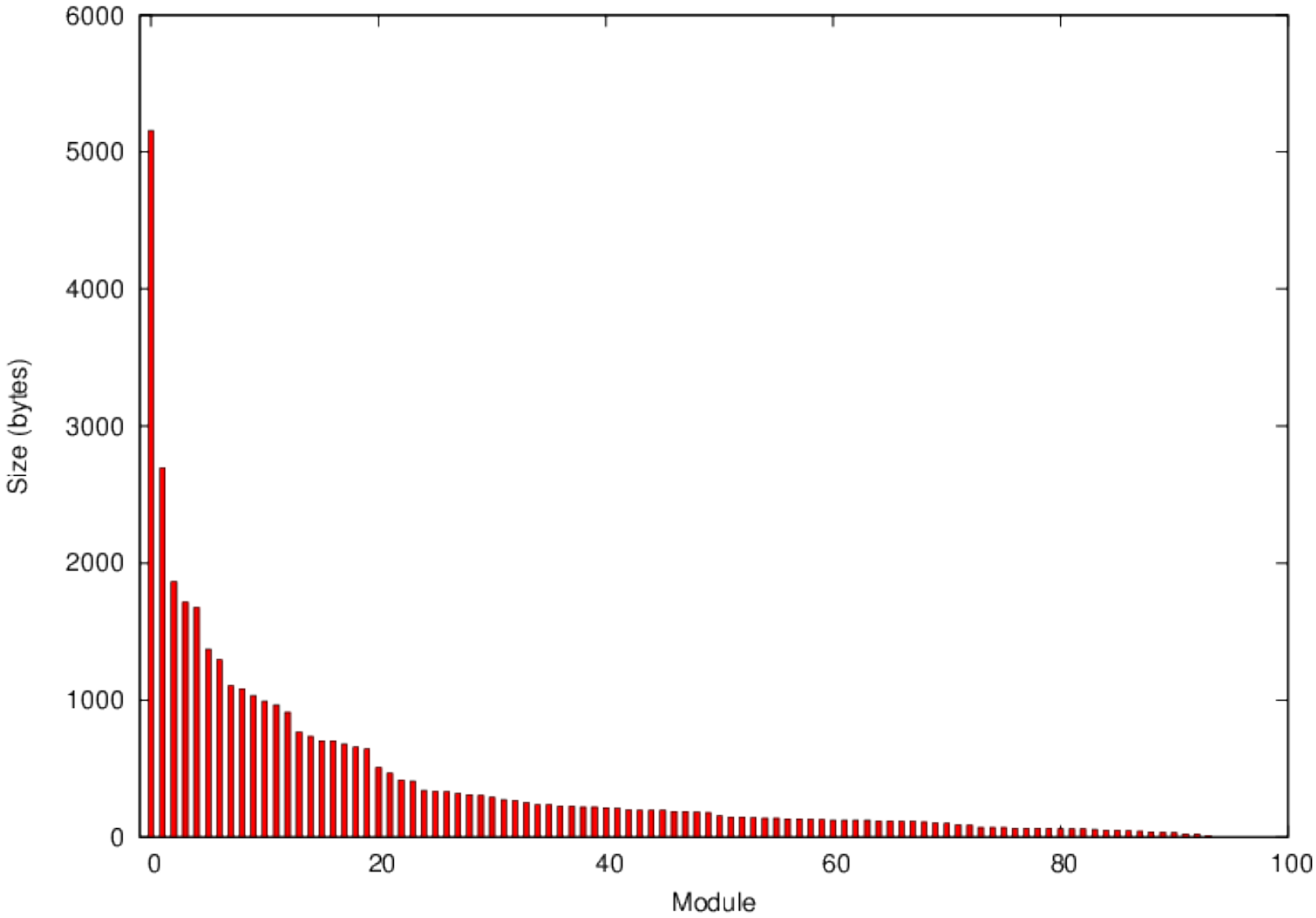
Why Modularization

- Growth management
- Software quality improvement
- Facilitating add-ons
- Separate third party libraries
- Optional Components

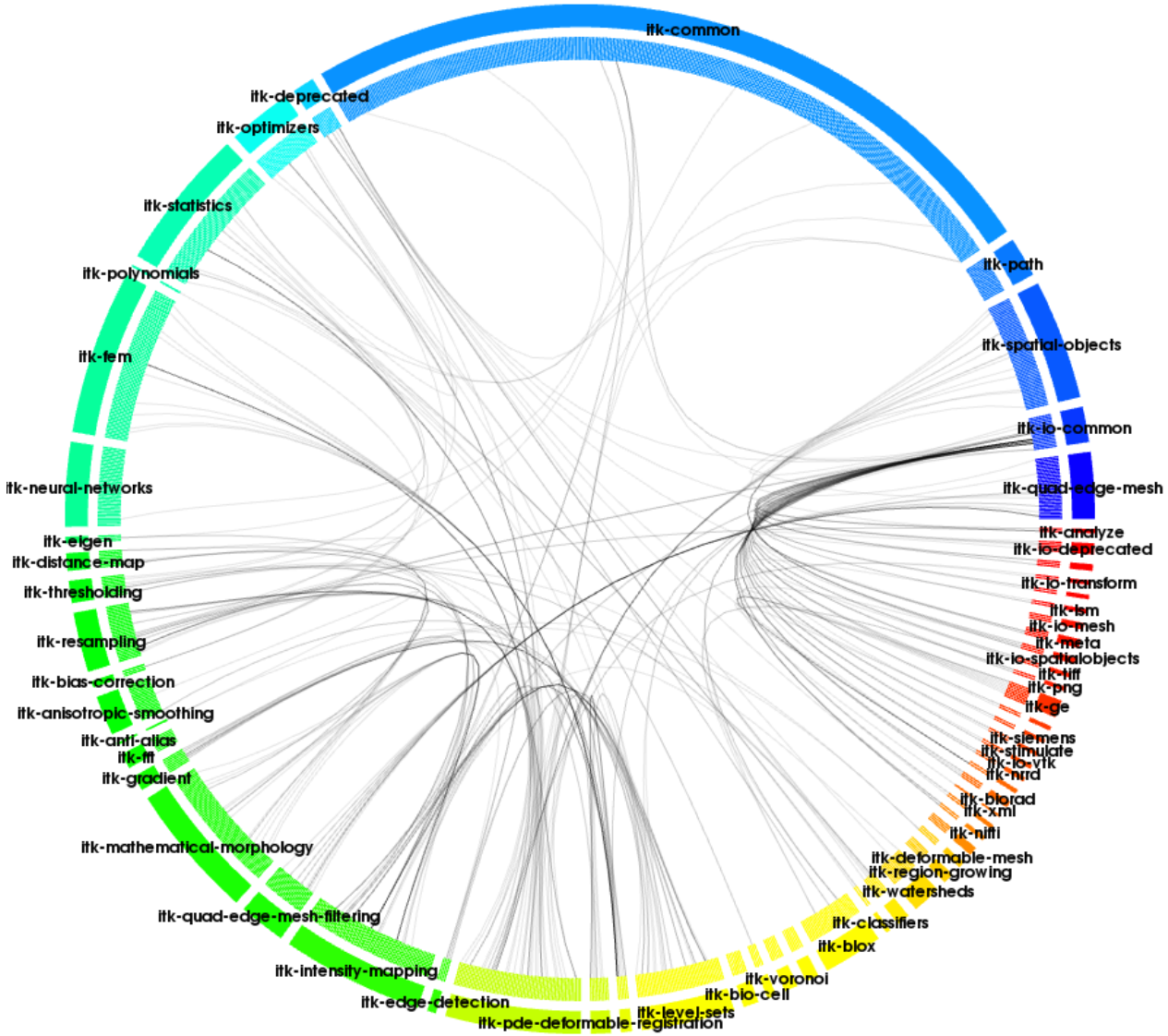
Module sizes



Module sizes (no third party modules)



Visualize dependencies among modules



ITK Module Grouping

- 12 groups, 107 modules and counting

ITK Module Grouping

- 12 groups, 107 modules and counting
- Grouped by module functionality

ITK Module Grouping

- 12 groups, 107 modules and counting
- Grouped by module functionality

Bridge	External	IO	Registration
Compatibility	Filtering	Nonunit	Segmentation
Core	GPU	Numerics	ThirdParty

How to add a module into ITK

1. Decide where to put the module

How to add a module into ITK

1. Decide where to put the module
2. Create the CMake setup

How to add a module into ITK

1. Decide where to put the module
2. Create the CMake setup
3. Add its content

Modularization checklist

- **Module categorization**: module group, module name

Modularization checklist

- **Module categorization**:
  - Group is specified by the directory

Modularization checklist

- **Module categorization**:
  - Group is specified by the directory
  - Module name is specified by directory, `itk-module.cmake`

Modularization checklist

- **Module categorization**: module group, module name
- **Directory hierarchy**: `include`, `src`, `test`

Modularization checklist

- **Module categorization**: module group, module name
- **Directory hierarchy**:
  - **include**: Headers defining the API, `*.hxx` template implementations

Modularization checklist

- **Module categorization**: module group, module name
- **Directory hierarchy**:
  - **include**: Headers defining the API, `*.hxx` template implementations
  - **src**: `*.cxx` implementations, create a library

Modularization checklist

- **Module categorization**: module group, module name
- **Directory hierarchy**:
  - **include**: Headers defining the API, `*.hxx` template implementations
  - **src**: `*.cxx` implementations, create a library
  - **test**: Module unit tests

Modularization checklist

- **Module categorization**: module group, module name
- **Directory hierarchy**: `include`, `src`, `test`
- **CMakeLists.txt** top of module, `src`, `test`

Modularization checklist

- **Module categorization**: module group, module name
- **Directory hierarchy**: `include`, `src`, `test`
- **CMakeLists.txt**
  - **top of module**: `project()`, `itk_module_impl()`, `set(ITKFoo_LIBRARIES ITKFoo)`

Modularization checklist

- **Module categorization**: module group, module name
- **Directory hierarchy**: `include`, `src`, `test`
- **CMakeLists.txt**
  - **top of module**: `project()`, `itk_module_impl()`, `set(ITKFoo_LIBRARIES ITKFoo)`
  - **src**: `add_library()`, `target_link_library()`,`itk_module_target()`

Modularization checklist

- **Module categorization**: module group, module name
- **Directory hierarchy**: `include`, `src`, `test`
- **CMakeLists.txt**
  - **top of module**: `project()`, `itk_module_impl()`, `set(ITKFoo_LIBRARIES ITKFoo)`
  - **src**: `add_library()`, `target_link_library()`,`itk_module_target()`
  - **test**: `itk_module_test()`, `itk_add_test()`

Modularization checklist

- **Module categorization**: module group, module name
- **Directory hierarchy**: `include`, `src`, `test`
- **CMakeLists.txt** top of module, `src`, `test`
- **Module dependencies and documentation** `itk-module.cmake`

External Modules

- Externally distributed ITK modules.
- Share the same modular structure as other interal modules.
- Needs to be manually downloaded to be used as a module.
- Facilitate code sharing and diverse development.
- Refer to: Lesion Sizing Toolkit Module (Modules/External/LesionSizingToolkit)

Remote Modules

- Modules distributed remotely.
- Can be fetched through cmake automatcially(Super build).
- With version control.
- Should have the same code quality standards as other internal ITK modules.

Exercise: ITKRAT

- Insight Journal article on itkRobustAutomaticThresholdImageFilter(RAT)
- `/Documents/ITKv4-TheNextGeneration-Tutorial/Exercises/Modulariz`
  `/IJ-submission-rat`
- Make into an External module: ITKRAT

More information on modularization

- Details about ITK Modularization can be found at Wiki Page.

Insight Journal

- Git Backend

- Git Backend
- Virtual Appliances

- Git Backend
- Virtual Appliances
- CDE

Insight Journal

- Git Backend
- Virtual Appliances
- CDE
- Reproducible Research

Migration Guide

http://itk.org/migrationv4

Live Long
and
Prosper !