ITK - The Next Generation

Welcoming and training new ITK Community Members

The University of Wisconsin-Madison

January 2013

Virtual Machine Preparation I

Virtual Machines Preparation

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

- VirtualBoxInstallers
  - VirtualBox-4.2.6-82870-OSX.dmg (Mac)
  - VirtualBox-4.2.6-82870-Win.exe (Windows)
  - VirtualBox-4.2.6-82870-Linux_amd64.run (Linux-64bit)
  - VirtualBox-4.2.6-82870-Linux_x86.run (Linux-32bit)

Media Content

Directories and Files

- VirtualBoxInstallers
  - VirtualBox-4.2.6-82870-OSX.dmg (Mac)
  - VirtualBox-4.2.6-82870-Win.exe (Windows)
  - VirtualBox-4.2.6-82870-Linux_amd64.run (Linux-64bit)
  - VirtualBox-4.2.6-82870-Linux_x86.run (Linux-32bit)
- VirtualMachine
  - ITK-Madison-Jan-2013.ova

Install VirtualBox

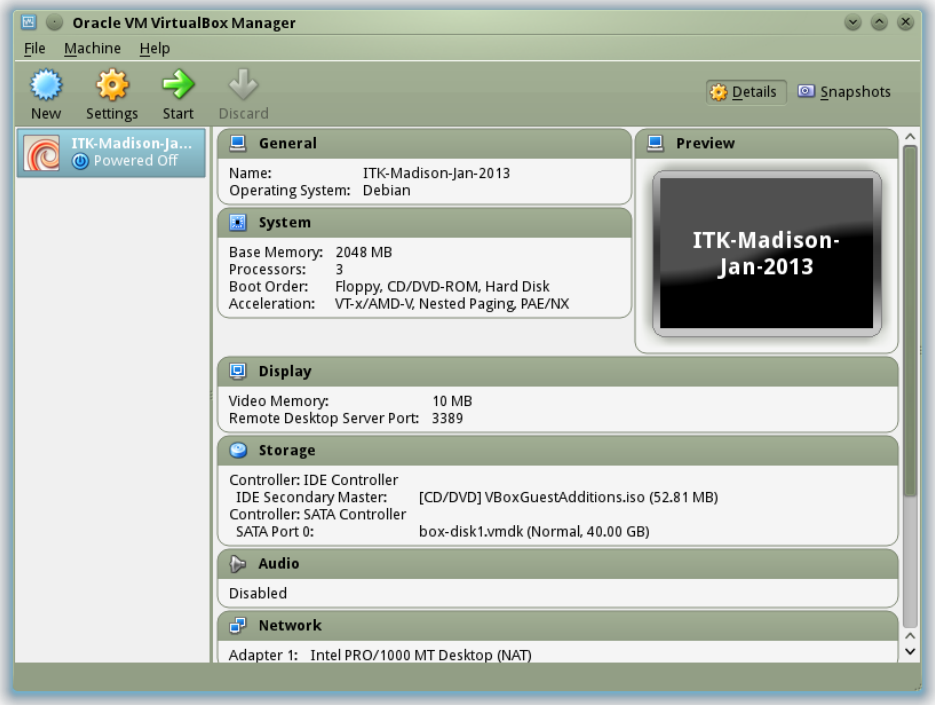
- Select the installer for your platform
- Run it

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 USB stick “VirtualMachine/ITK-Madison-2013.ova”
- A progress bar will appear, and when it finishes you should see:

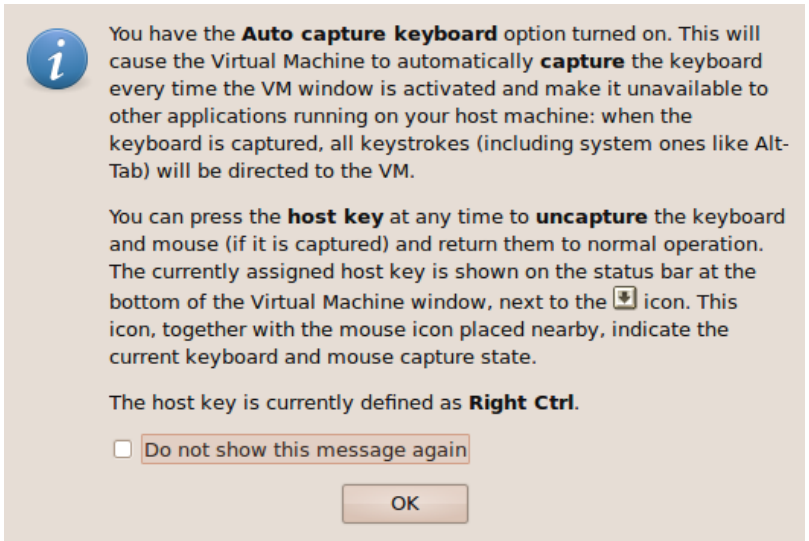


Introduction to ITK...

Virtual Machine Preparation II

Booting the Virtual Machine

- Click on the "ITK-Madison-Jan-2013" 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:

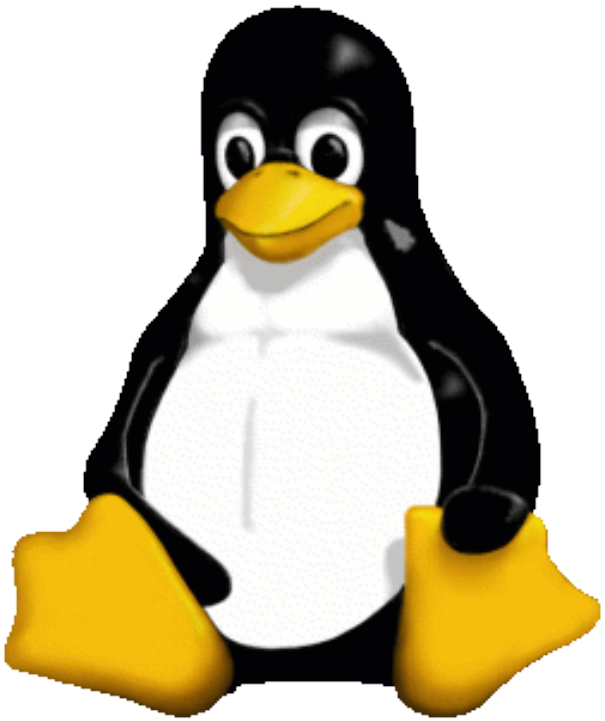
Booting the Virtual Machine

- Username: itk
- Password: itk

Your Virtual Machine is Ready !

Software Environment

Welcome to Debian GNU/Linux

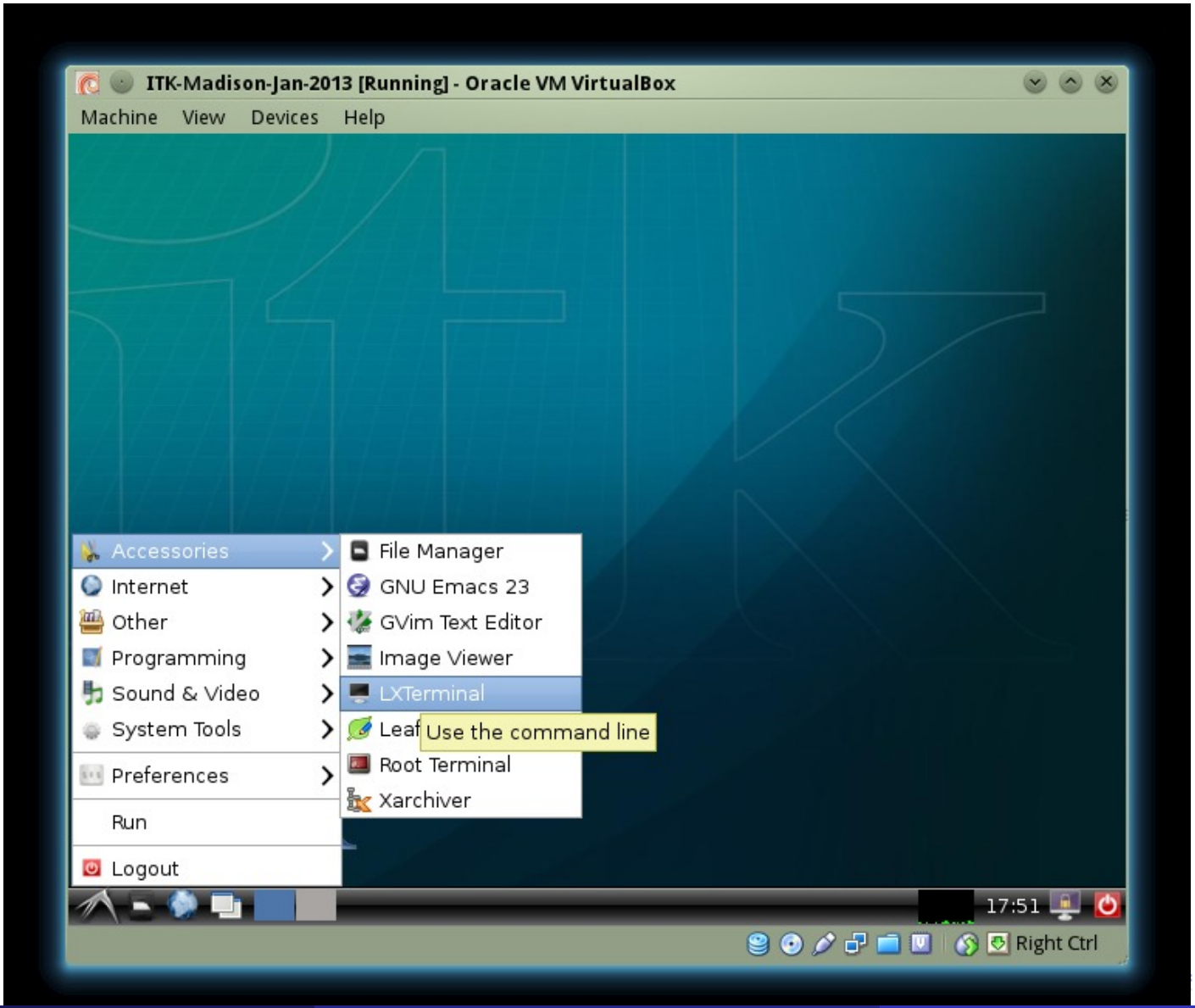


How to take the mouse out of the Virtual Machine

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

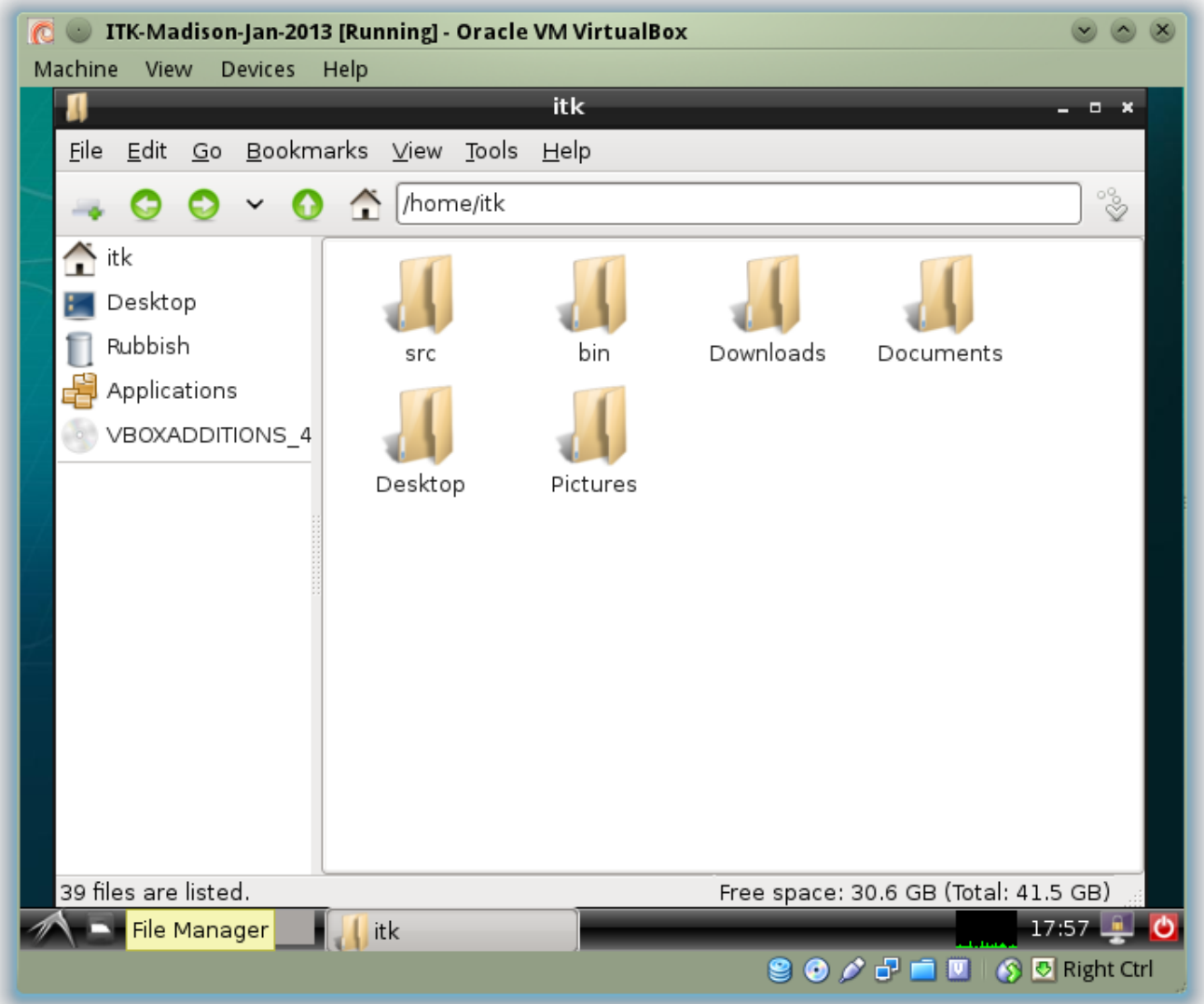
How to Open a Terminal - Menu in Lower Left Corner

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 ~/src/ITKv4-TheNextGeneration-Tutorial/Exercises
pwd
ls
pcmanfm .
```

Walk through the directories

- Find source code of exercises

```
cd ~/src/ITKv4-TheNextGeneration-Tutorial/Exercises
pwd
ls
pcmanfm .
```

- Find binary build of exercises

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

SimpleITK

- Hides templates

SimpleITK

- Hides templates
- Smarter IO

SimpleITK

- Hides templates
- Smarter IO
- Both procedural and object-oriented interfaces

SimpleITK

- Hides templates
- Smarter IO
- Both procedural and object-oriented interfaces
- Easier wrapping

SimpleITK

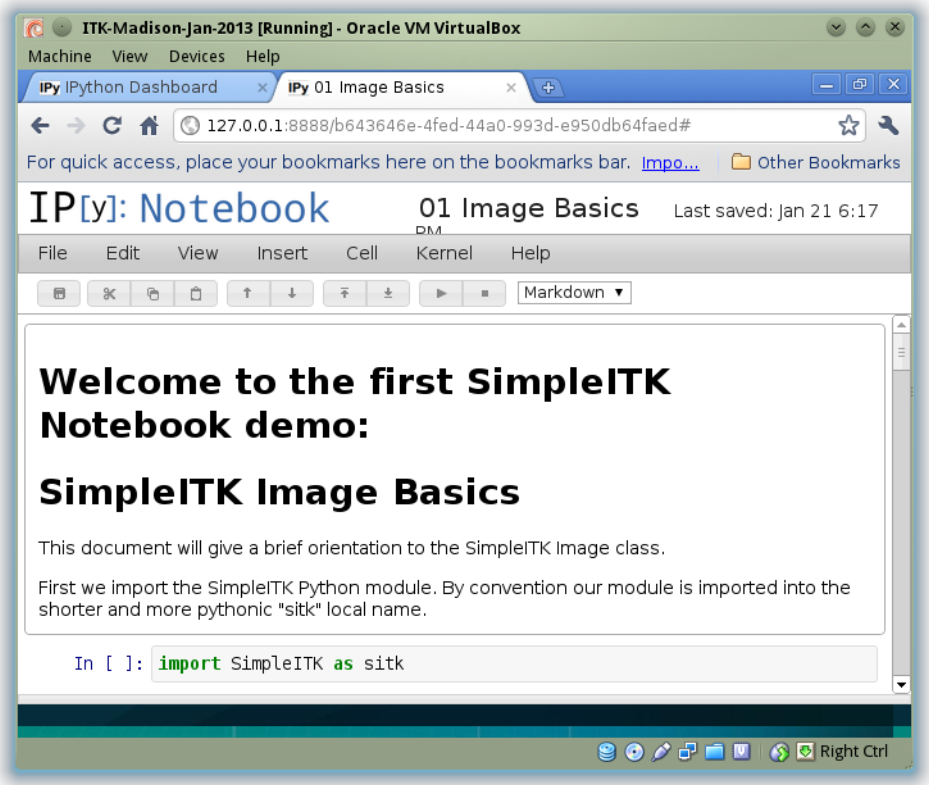
- Hides templates
- Smarter IO
- Both procedural and object-oriented interfaces
- Easier wrapping
- Binary distribution (`easy_install SimpleITK`)

How to run the example?

```
cd ~/src/SimpleITK-Notebooks

ipython notebook --pylab=inline
```

IPython Notebook Quickstart

- `Shift-Enter` to run a cell
- `Ctrl-m h` to show keyboard shortcuts



More information on SimpleITK

- http://simpleitk.org/
- http://www.itk.org/Wiki/ITK_Release_4/SimpleITK

Refactored Level Sets

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

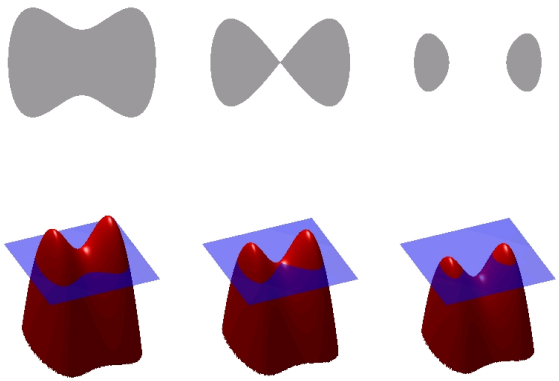
Arnaud Gelas Kishore Mosaliganti Sean Megason

Introduction

What is a level-set function?

Definition

- Implicit function $\phi : \Omega \to \mathbb{R}$
- 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

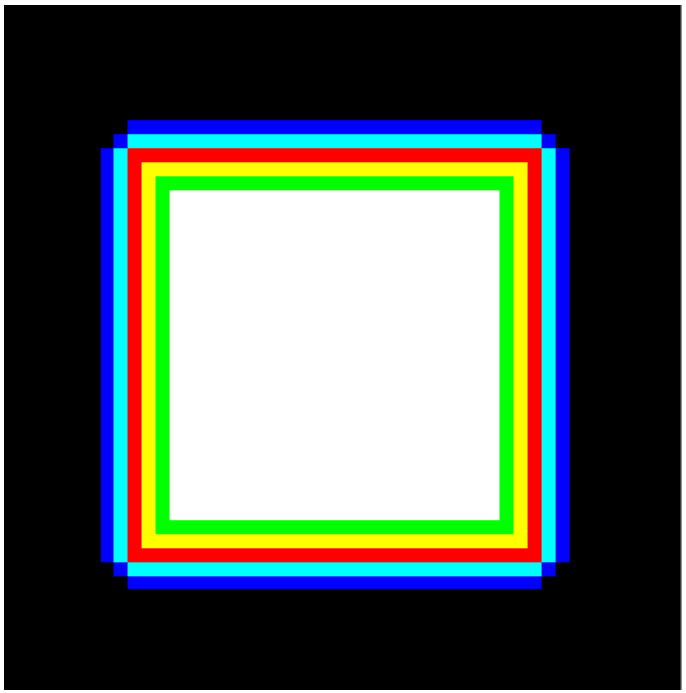
Discrete

- Dense
- Sparse

Level Sets Representation

Discrete

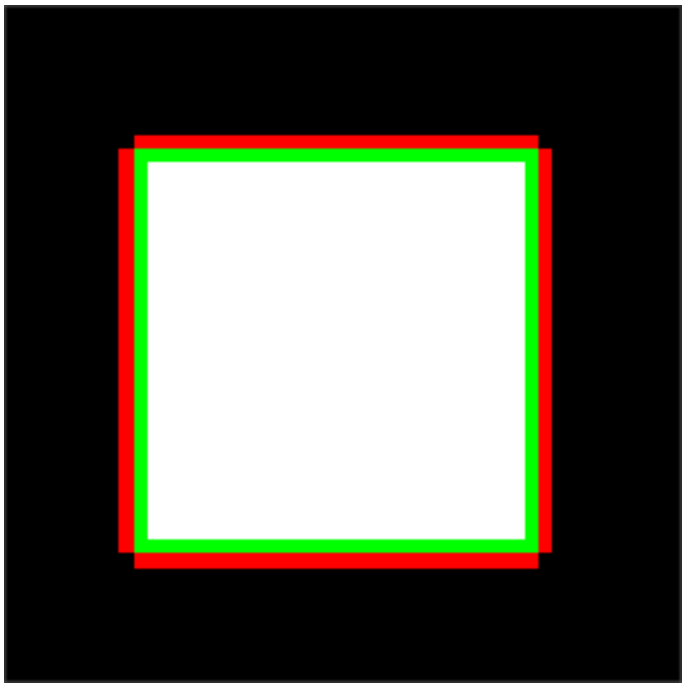
- Dense
- Sparse
  1. Whitaker



Level Sets Representation

Discrete

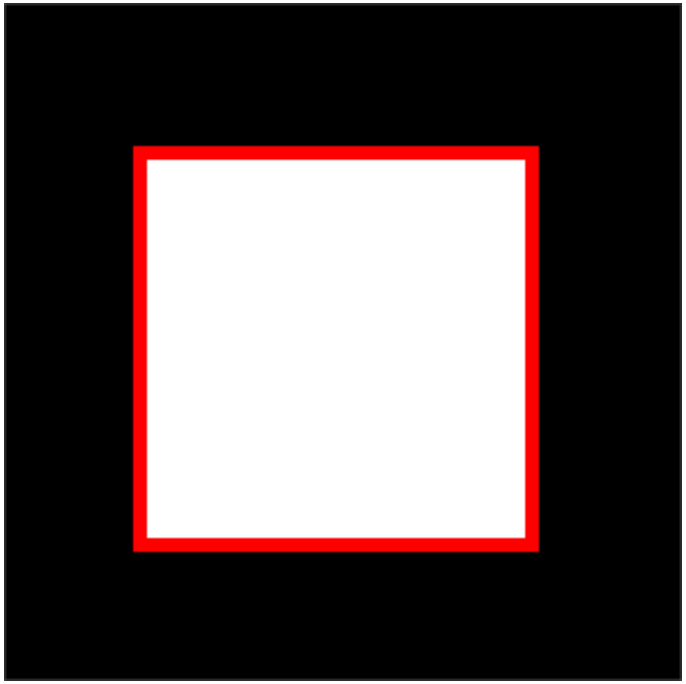
- Dense
- Sparse
  1. Whitaker
  2. Shi



Level Sets Representation

Discrete

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



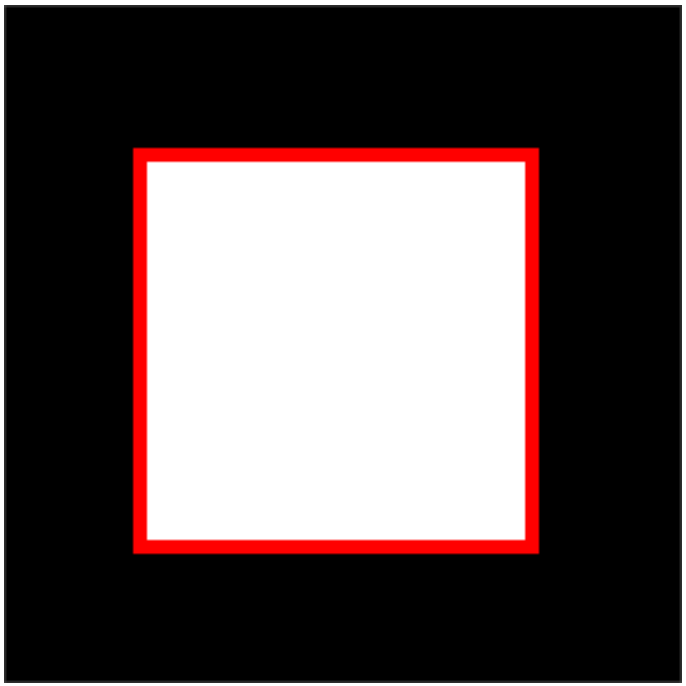
Level Sets Representation

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/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::LevelSetEquationTermContainer<
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::LevelSetEquationContainer< 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

Refactored Registration Framework

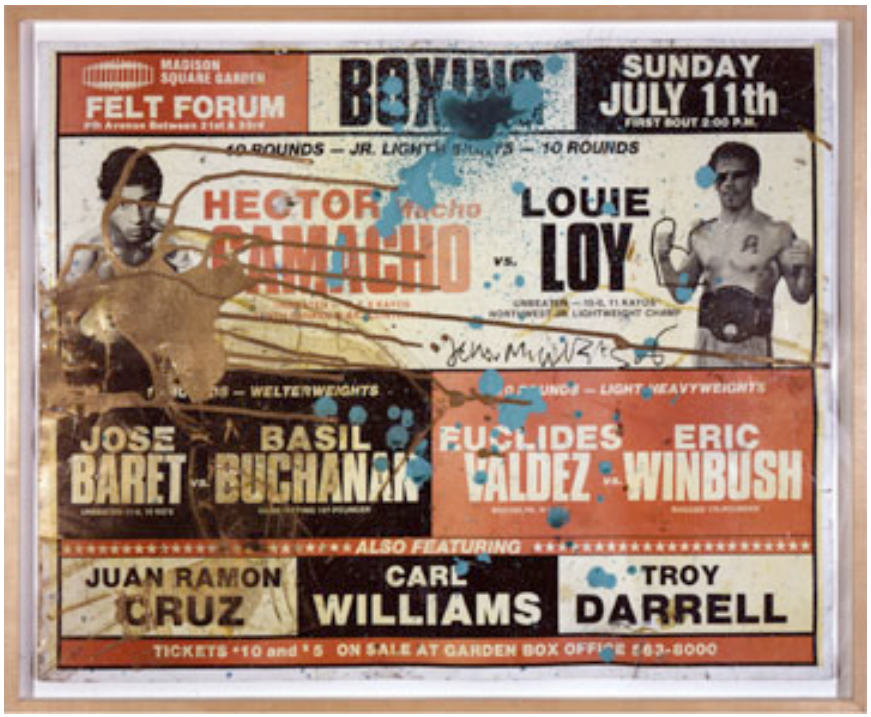
Insight Software Consortium

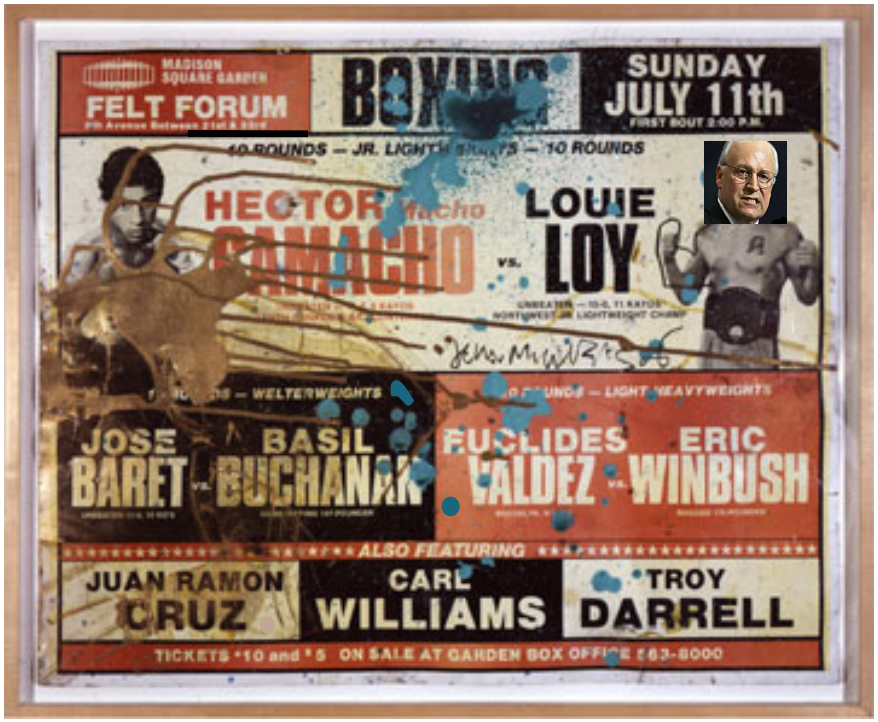
PICSL @ University of Pennsylvania

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

What is registration?

What is registration?





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

Image Origin and Spacing

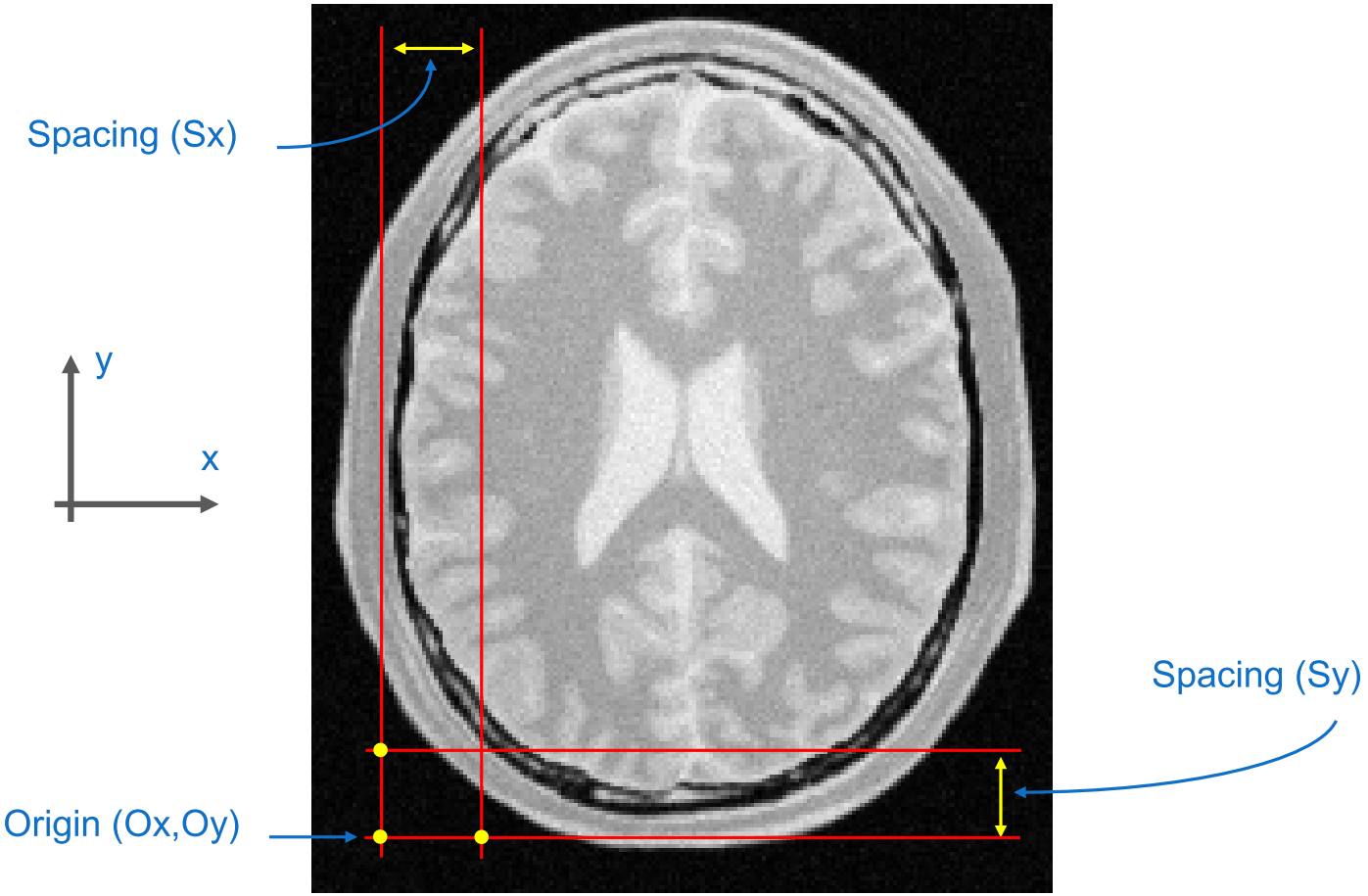


Image Sampling Grid

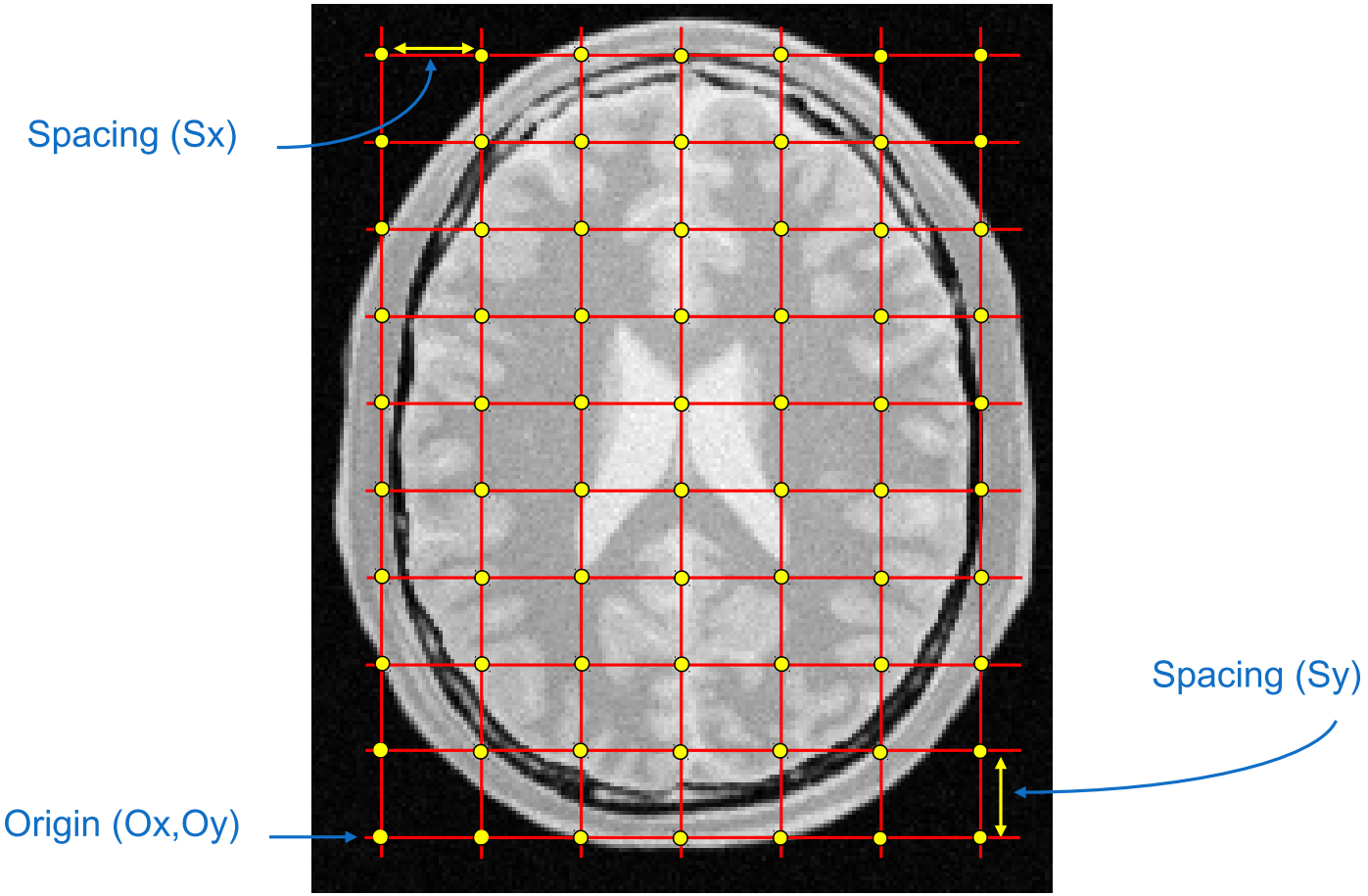


Image Pixel

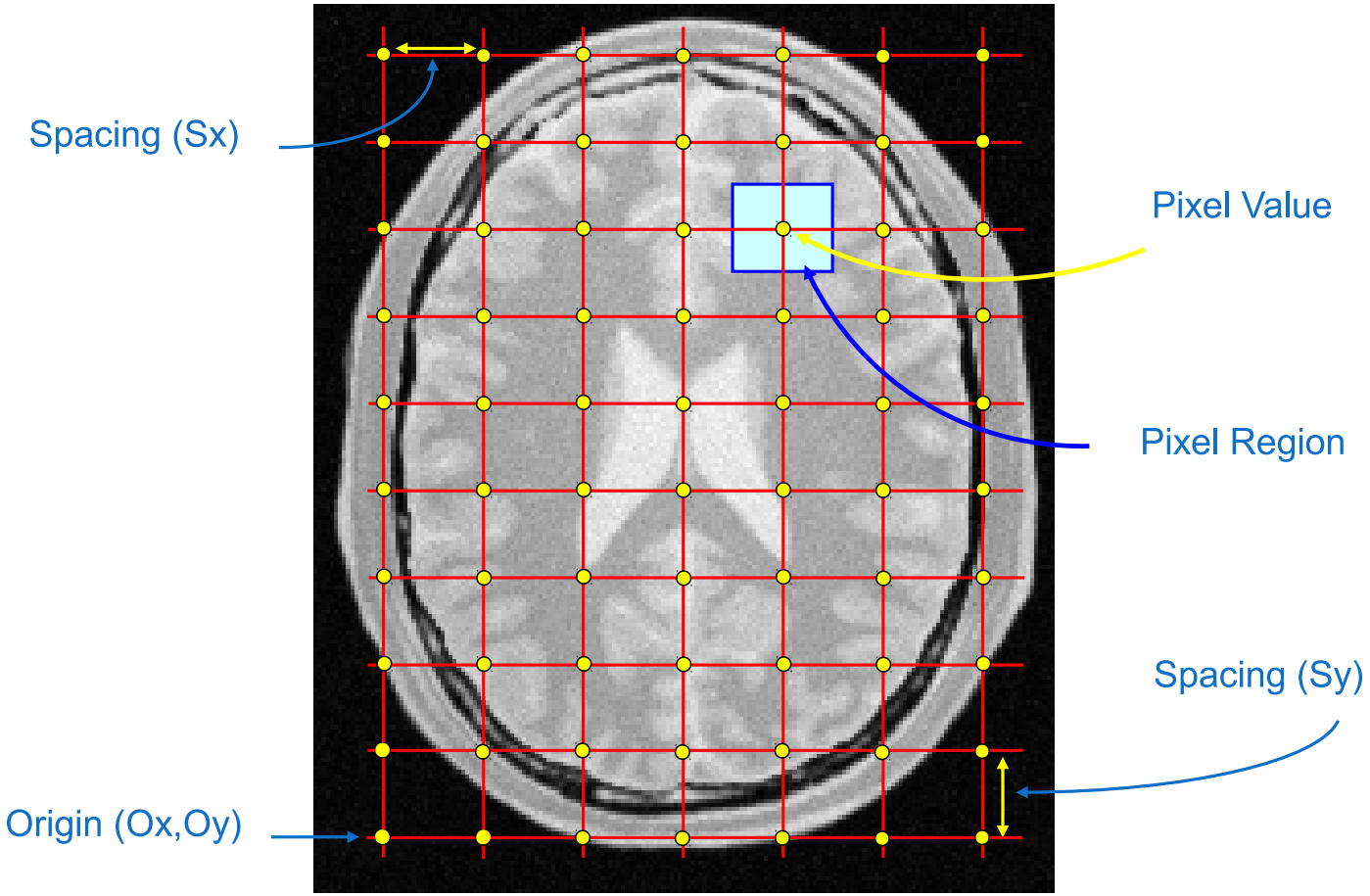


Image Indices

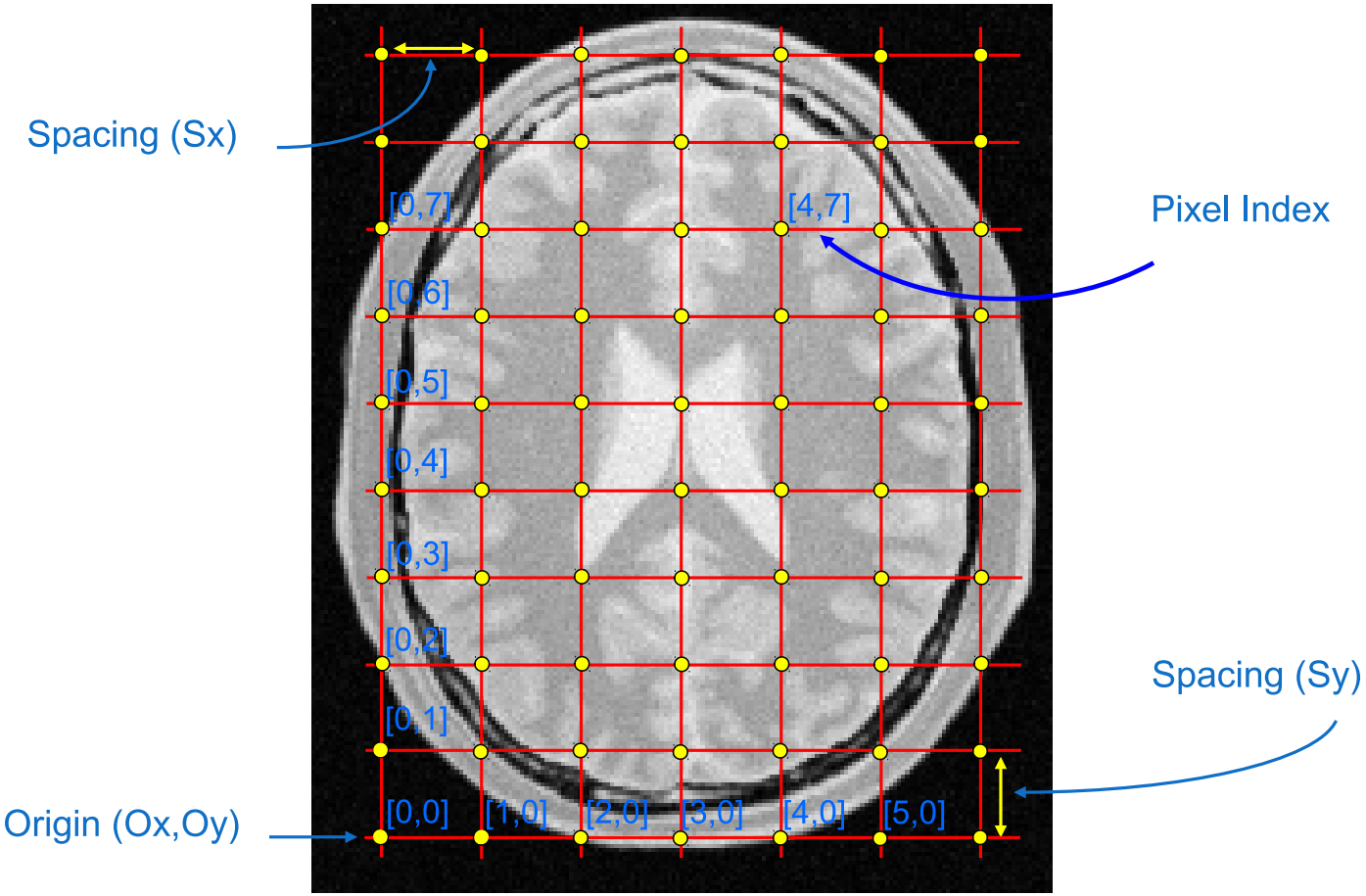


Image To Physical Coordinates

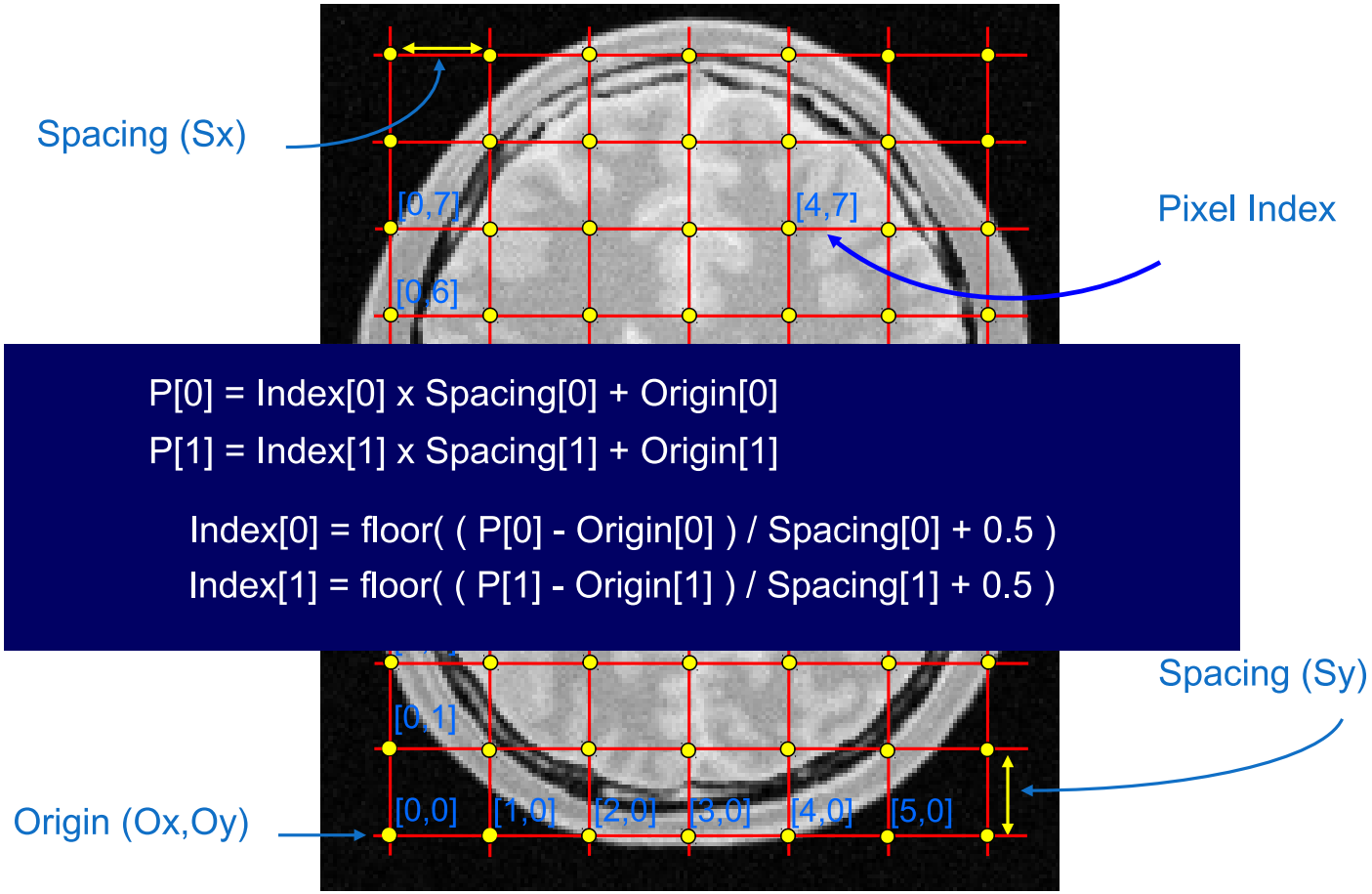


Image Region

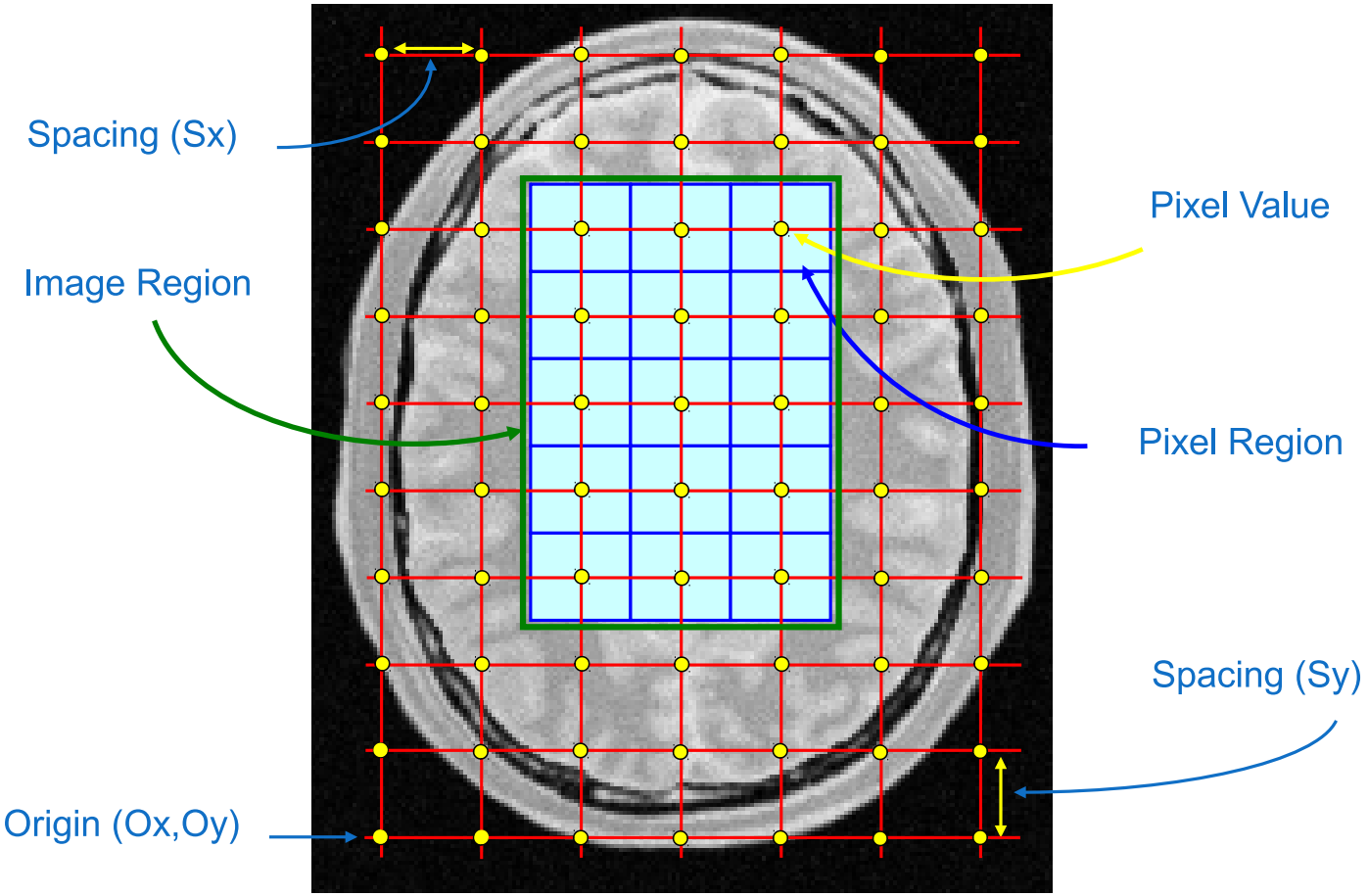


Image Region: Size

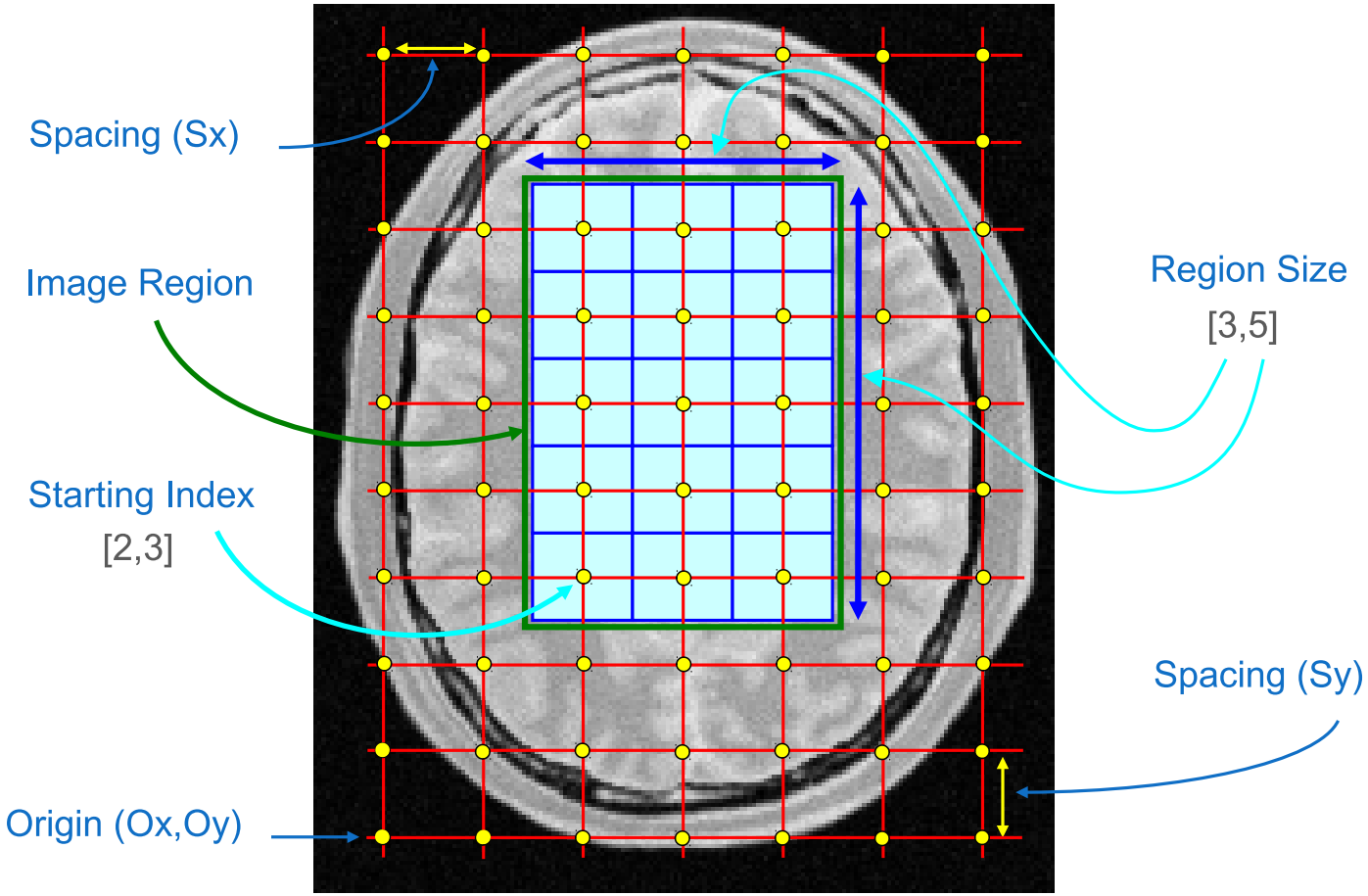
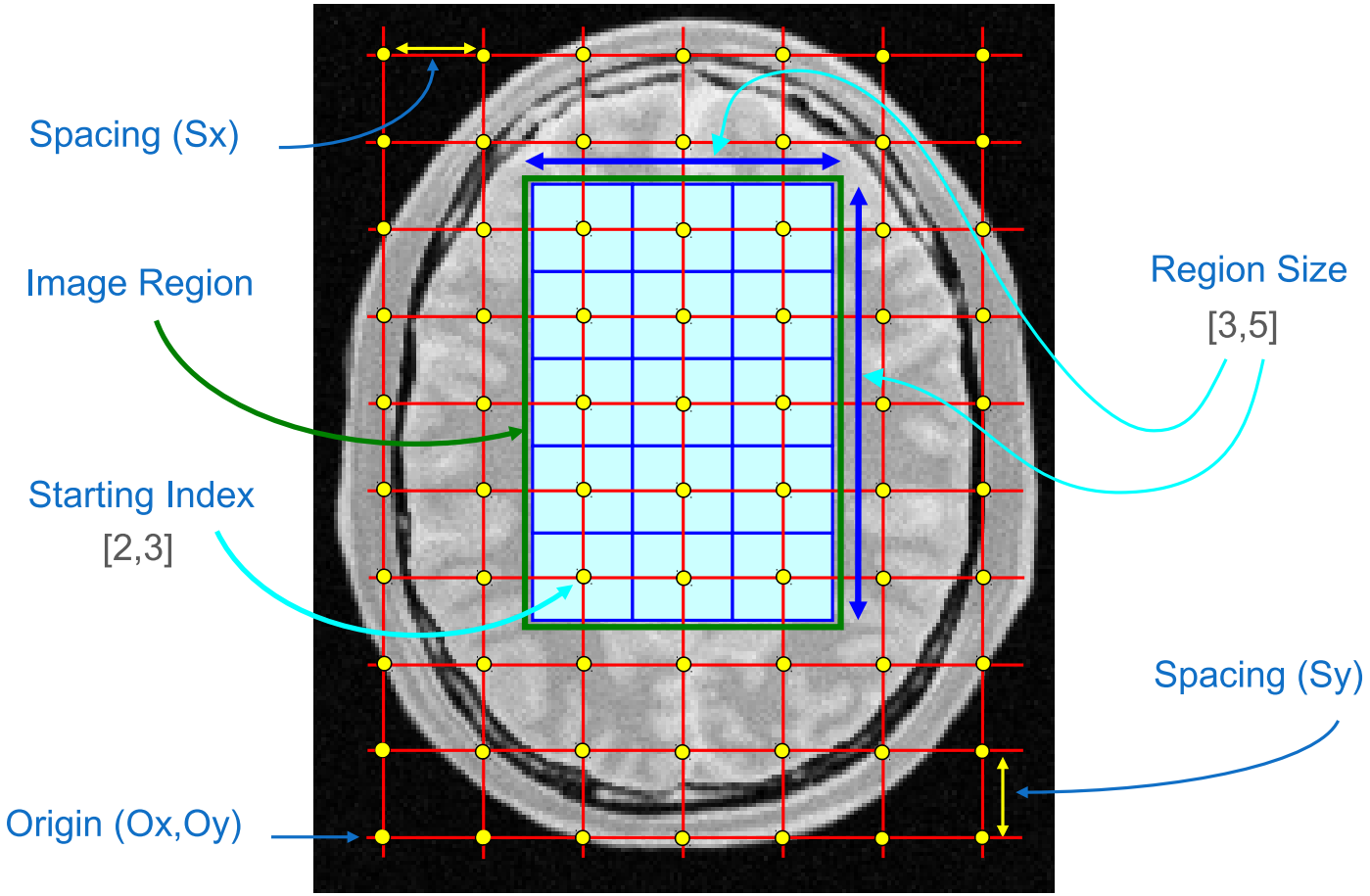
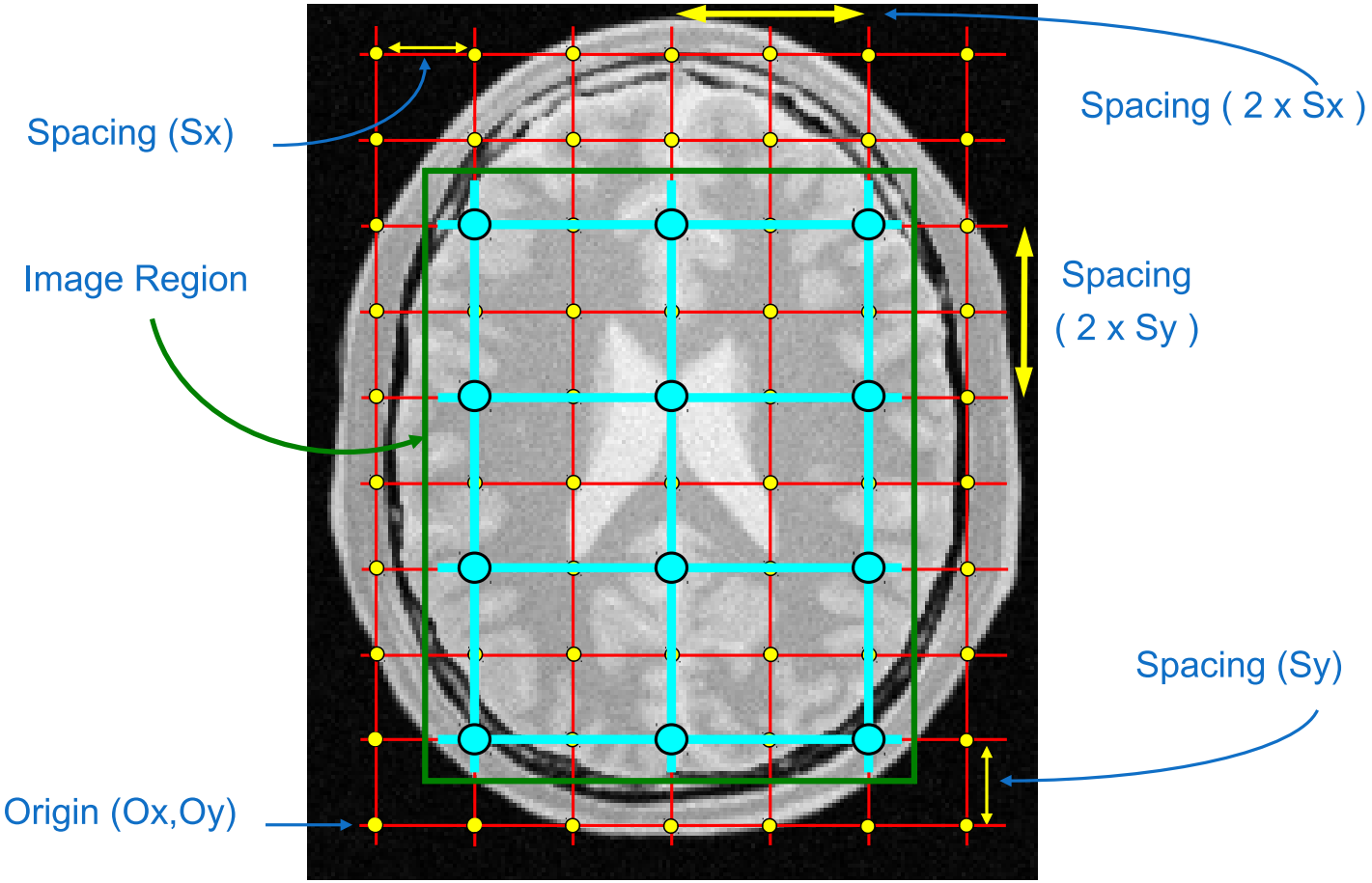


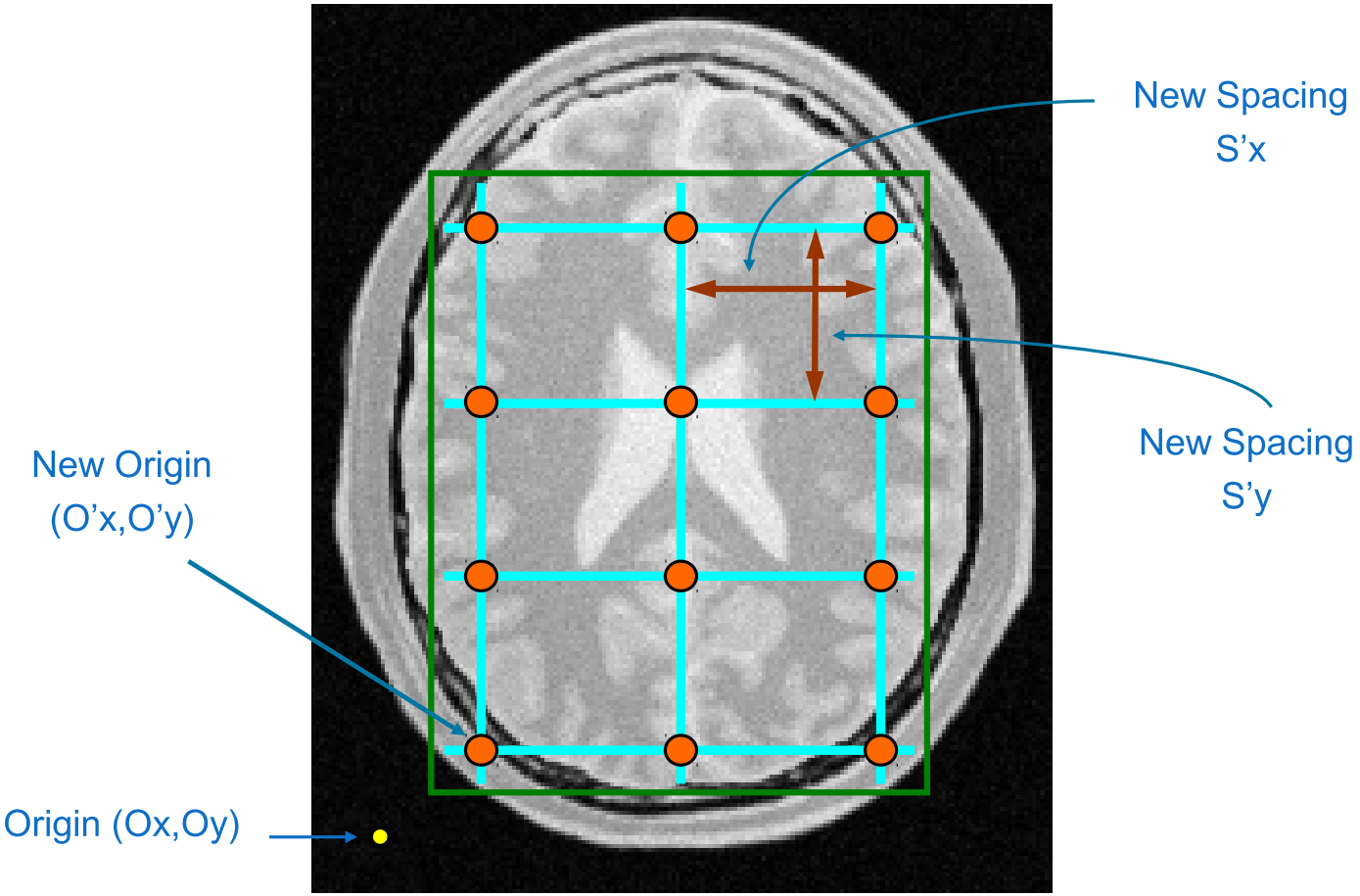
Image Region: Starting Index



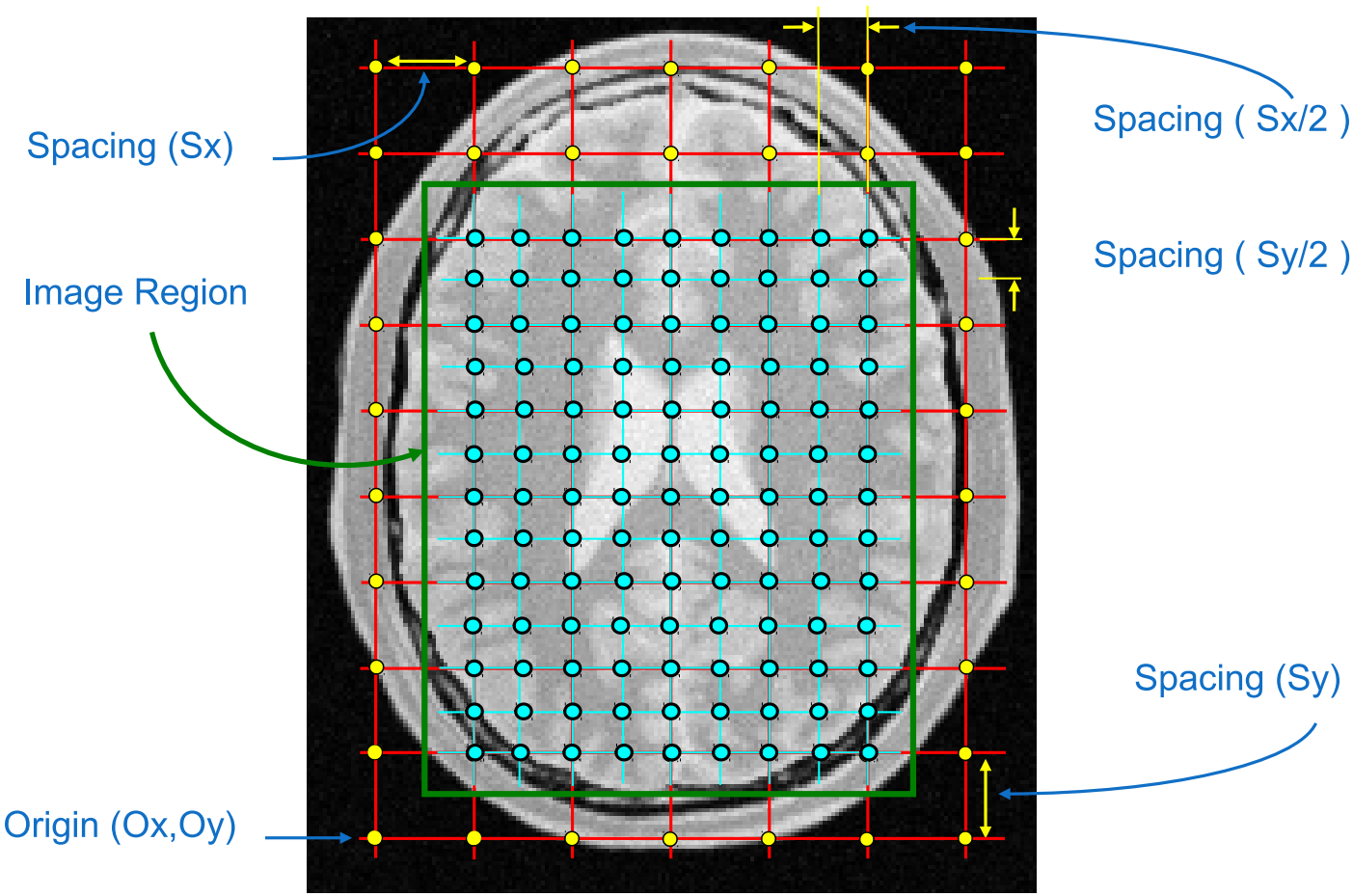
Sub-sampling by Half



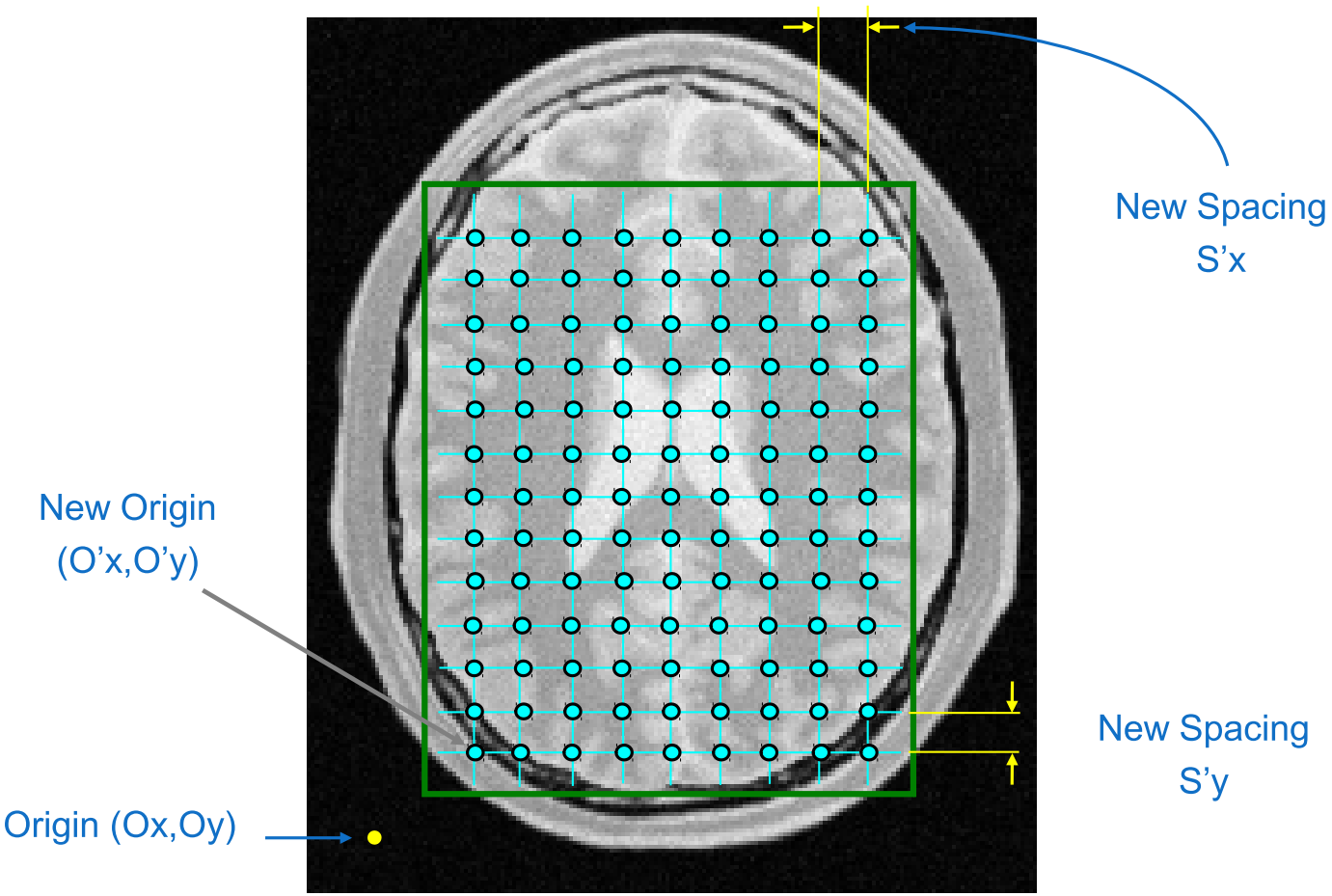
Sub-sampling by Half



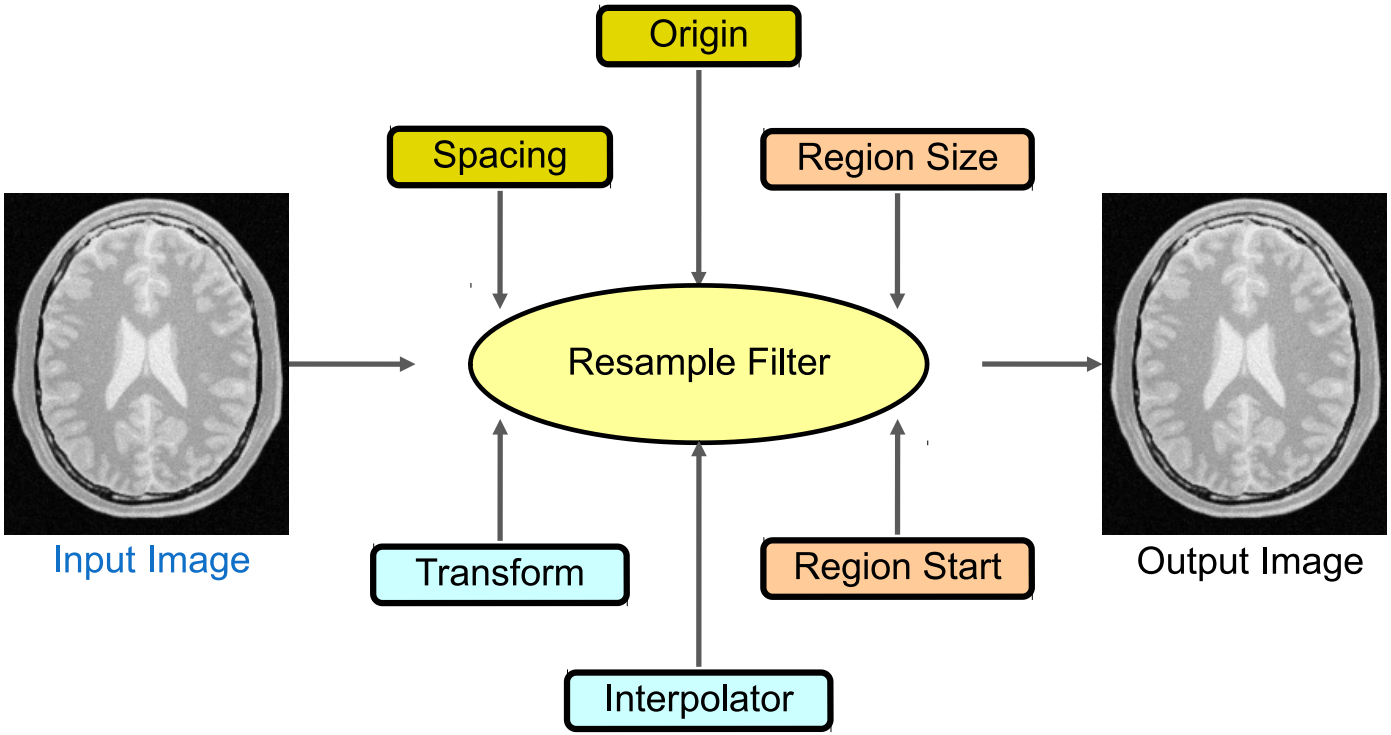
Super-sampling by Double



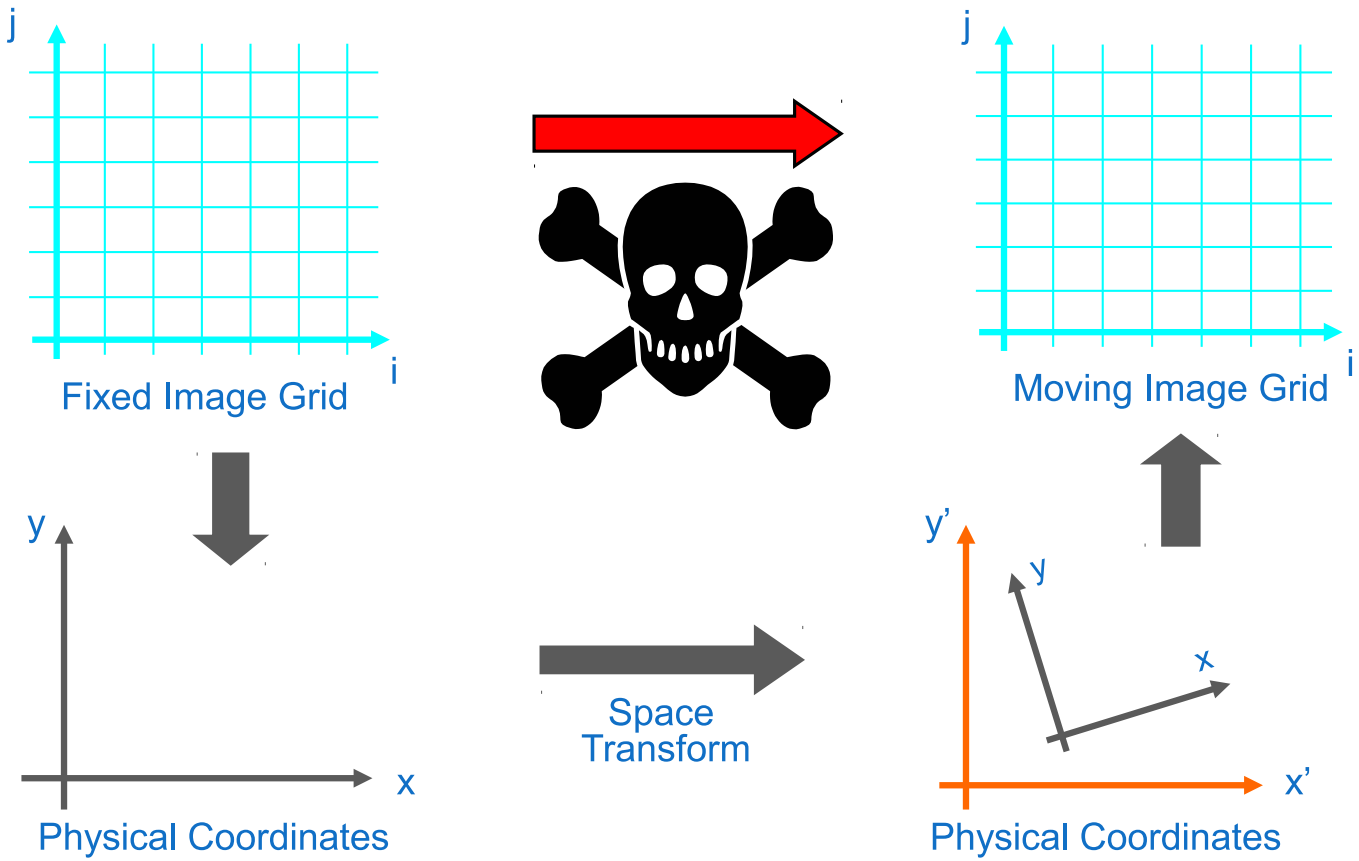
Super-sampling by Double



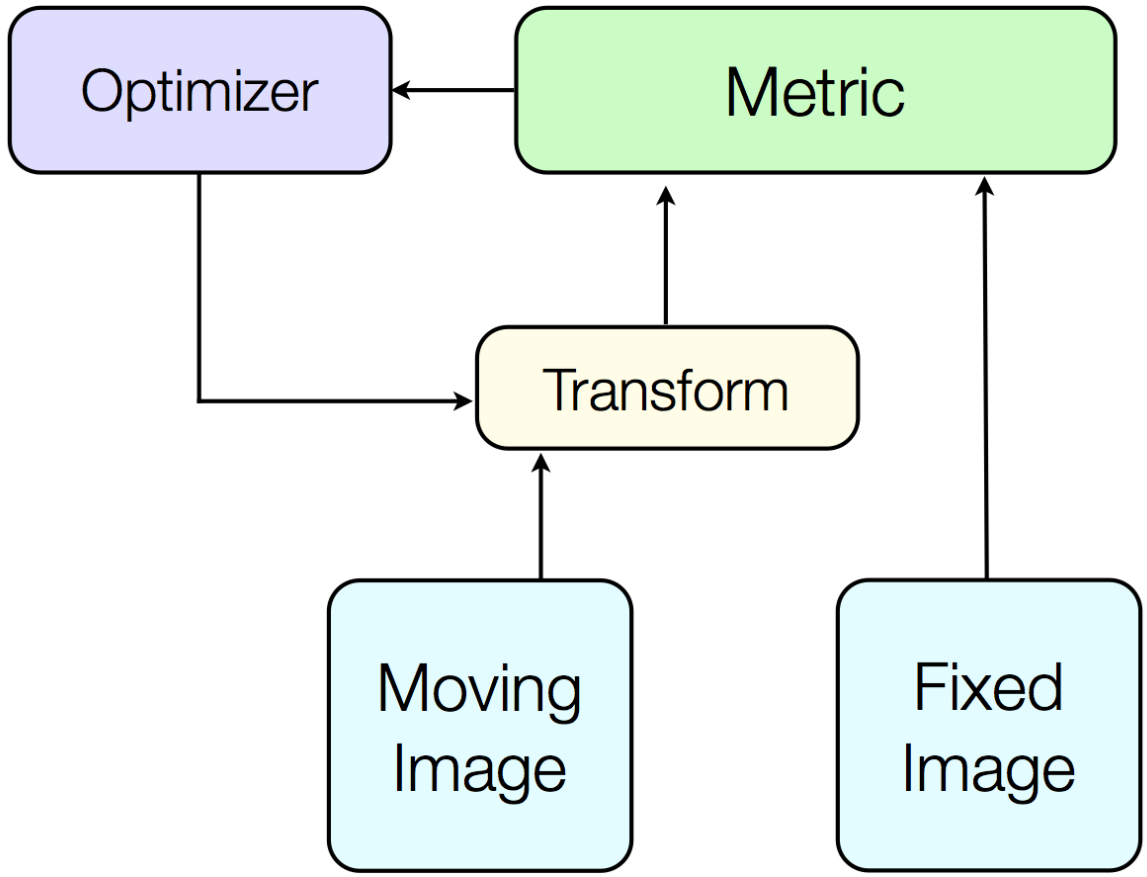
Resampling In ITK



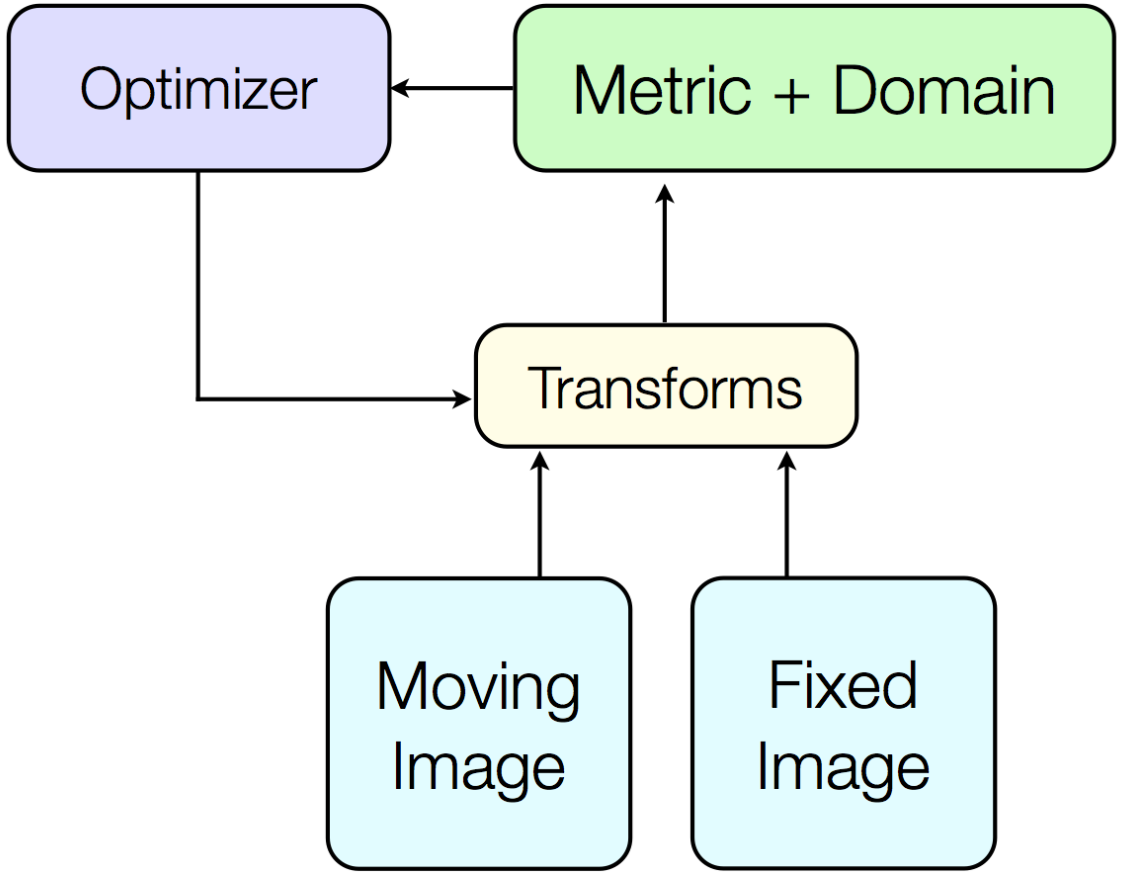
Coordinate System Conversions



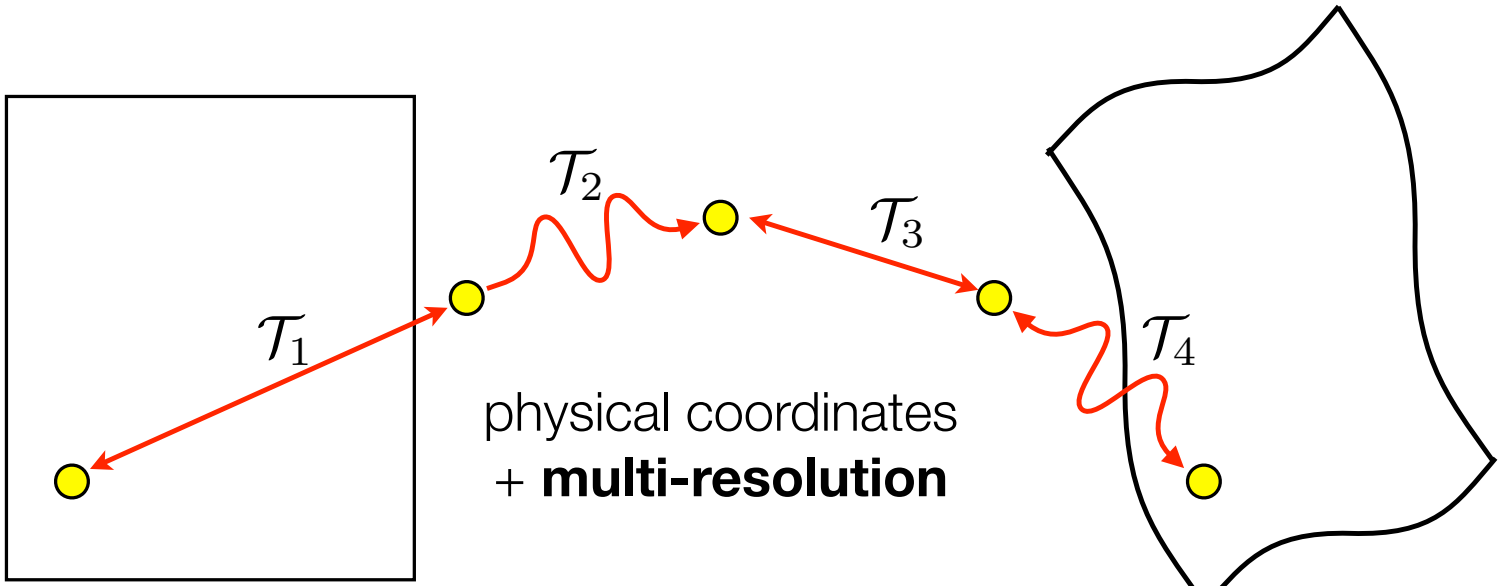
ITK v3 framework



ITK v4 framework



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.

Example Registration

Example: Perform a simple registration

```
leafpad ~/src/ITKv4-TheNextGeneration-Tutorial/\
Exercises/Registration/RegistrationExercises.cxx &

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

Example: Perform a simple registration

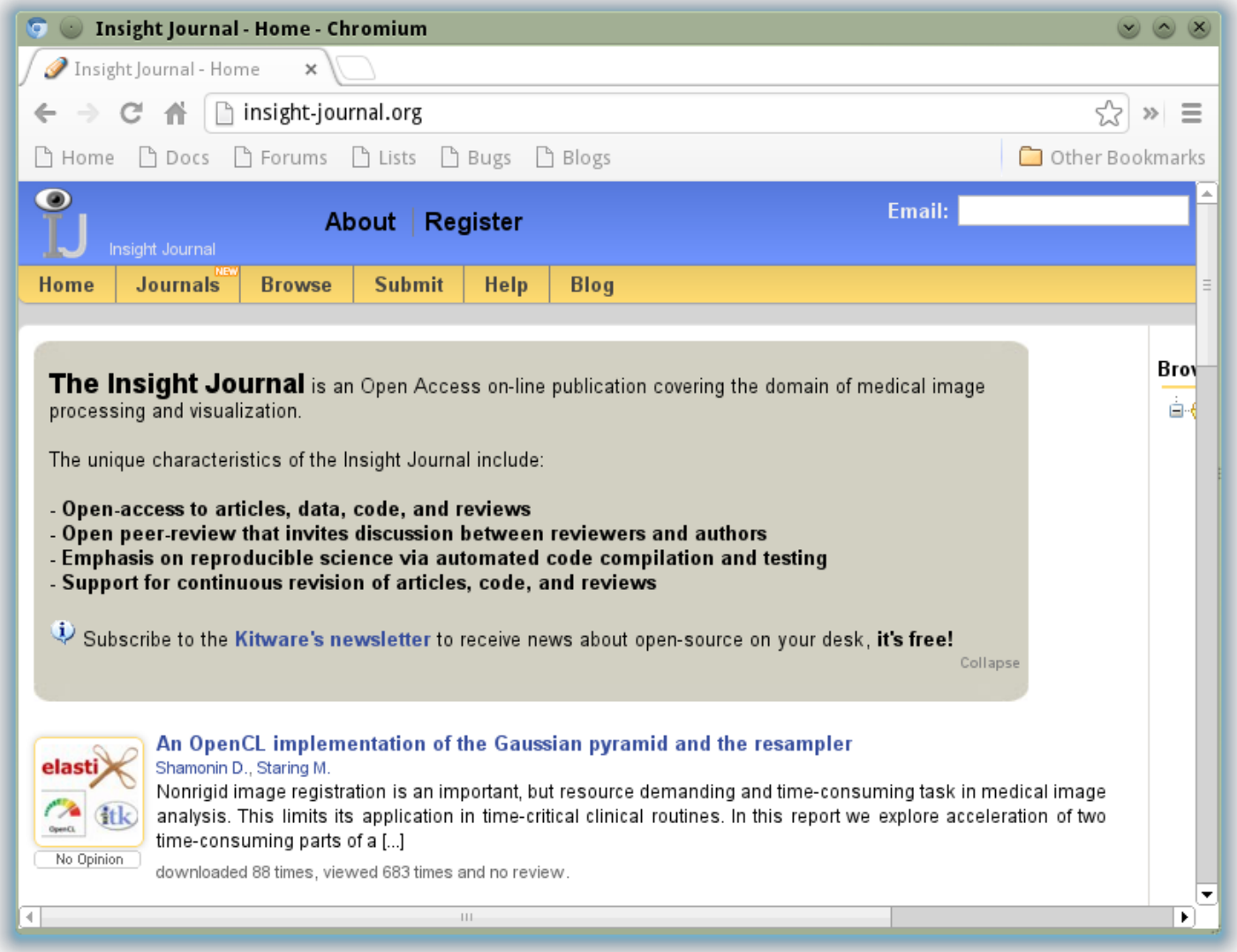
```
./bin/RegistrationExercises ~/data/BrainProtonDensitySliceBorc
  ~/data/BrainProtonDensitySliceR10X13Y17.png \
  ./Registered.png \
  3000

ImageViewer ./Registered.png &
```

Exercise: Do not scale the optimizer parameters

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

Insight Journal

insight-journal.org



- Git Backend

- Git Backend
- Virtual Appliances

- Git Backend
- Virtual Appliances
- CDE

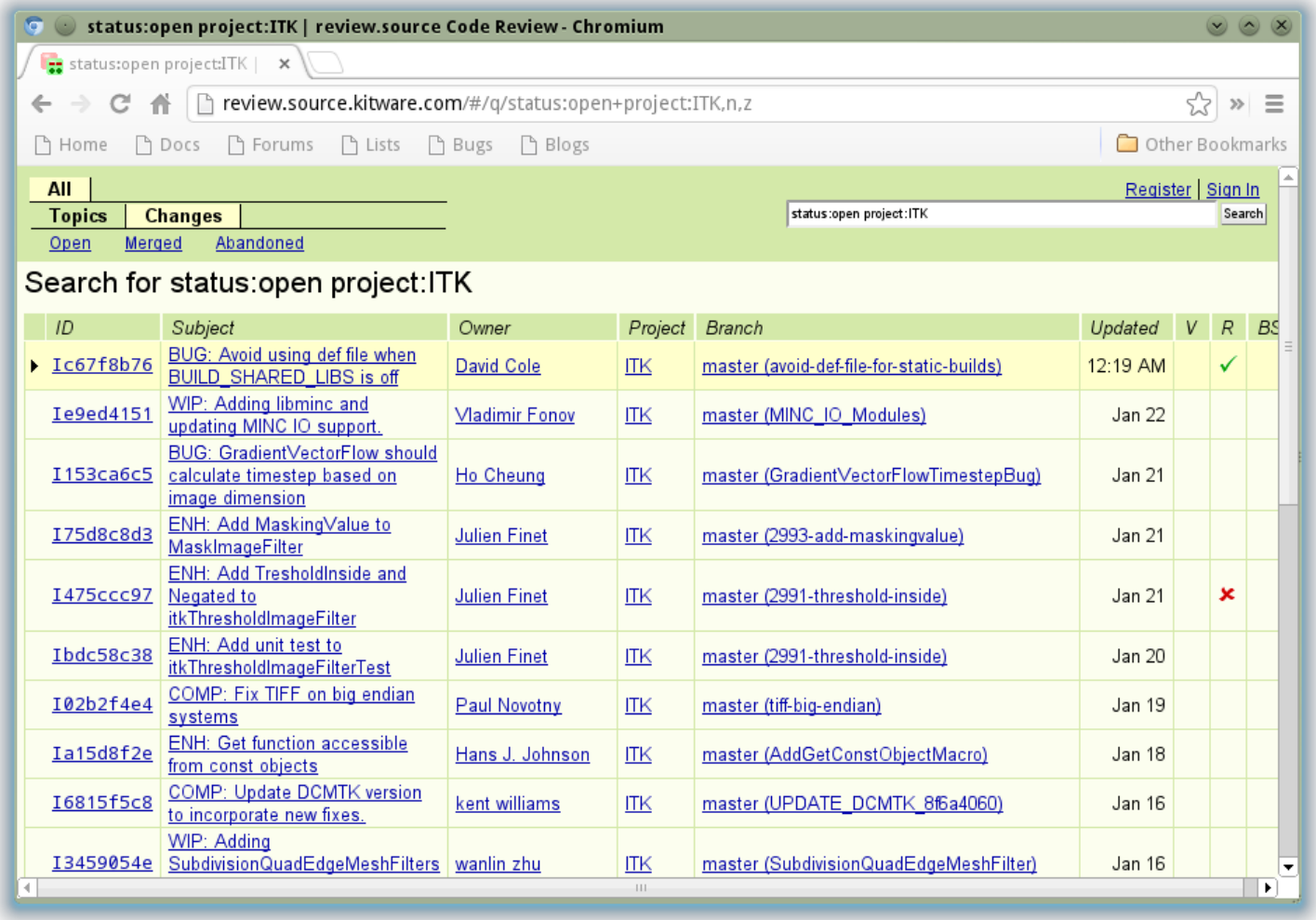
Insight Journal

- Git Backend
- Virtual Appliances
- CDE
- Reproducible Research

Gerrit Code Review

- Submit patches
- Review patches
- `http://www.itk.org/Wiki/ITK/Git/Develop`
- `http://review.source.kitware.com`

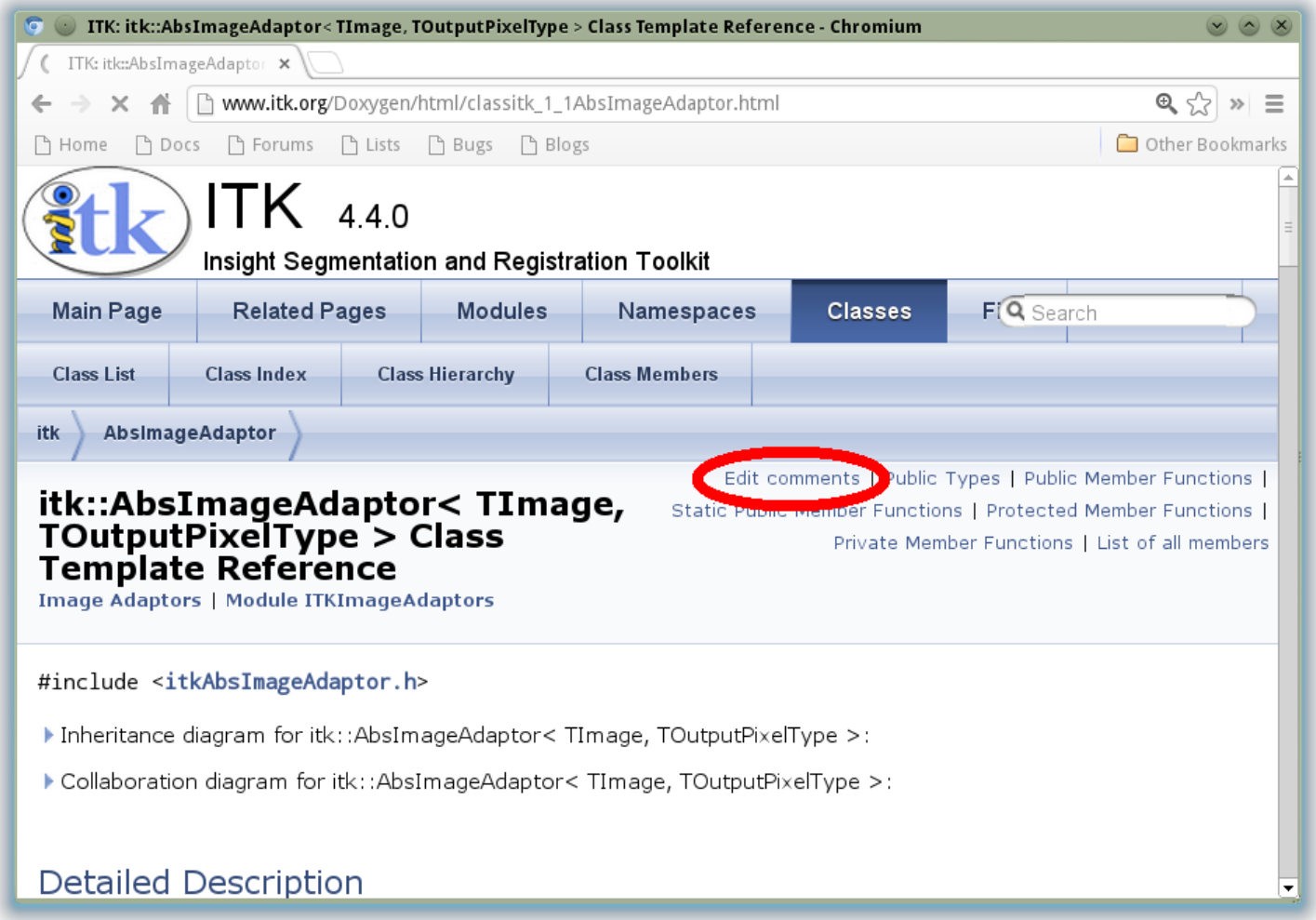
review.source.kitware.com



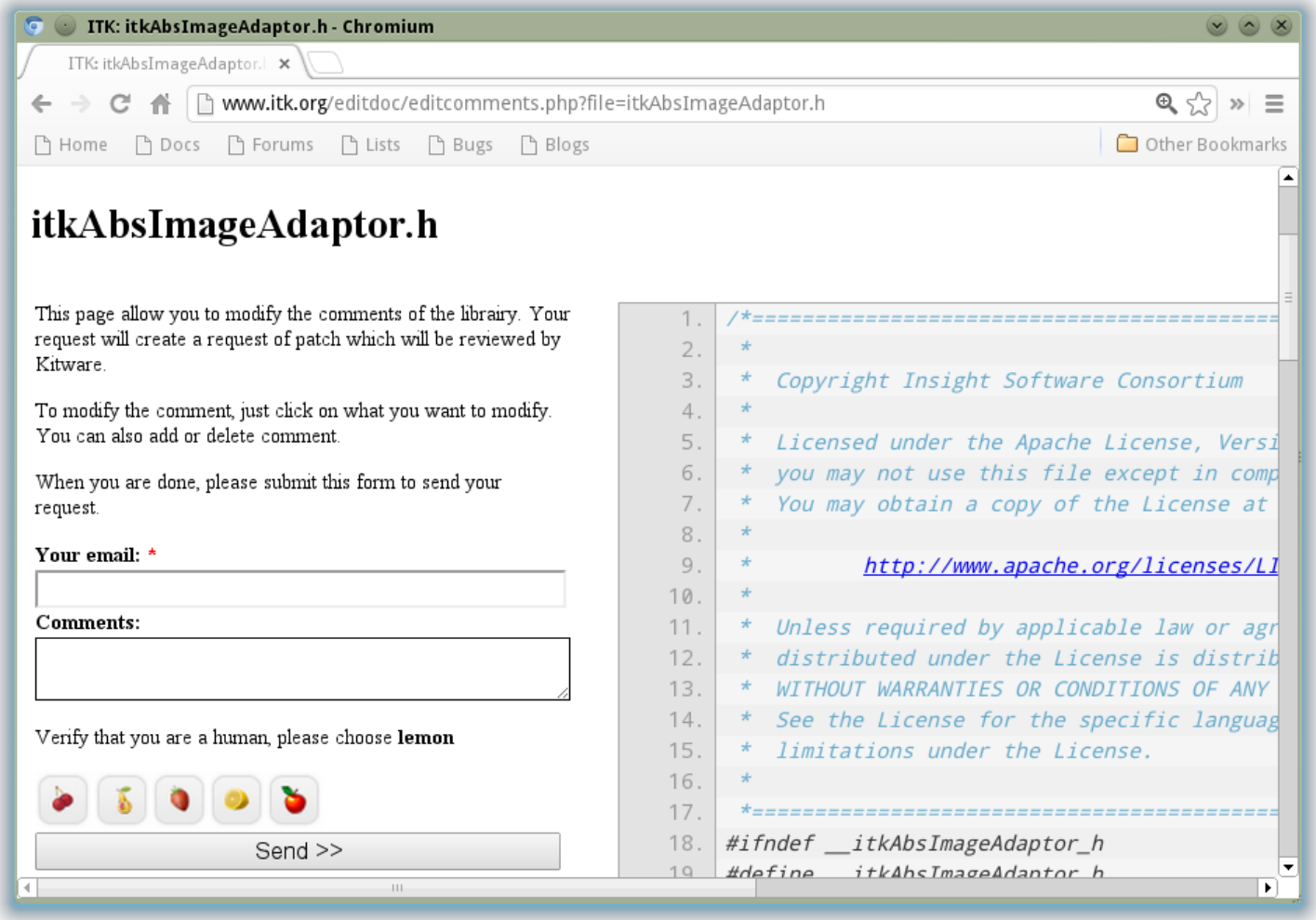
Mailing Lists

- http://www.itk.org/mailman/listinfo/insight-users
- http://www.itk.org/mailman/listinfo/insight-developers

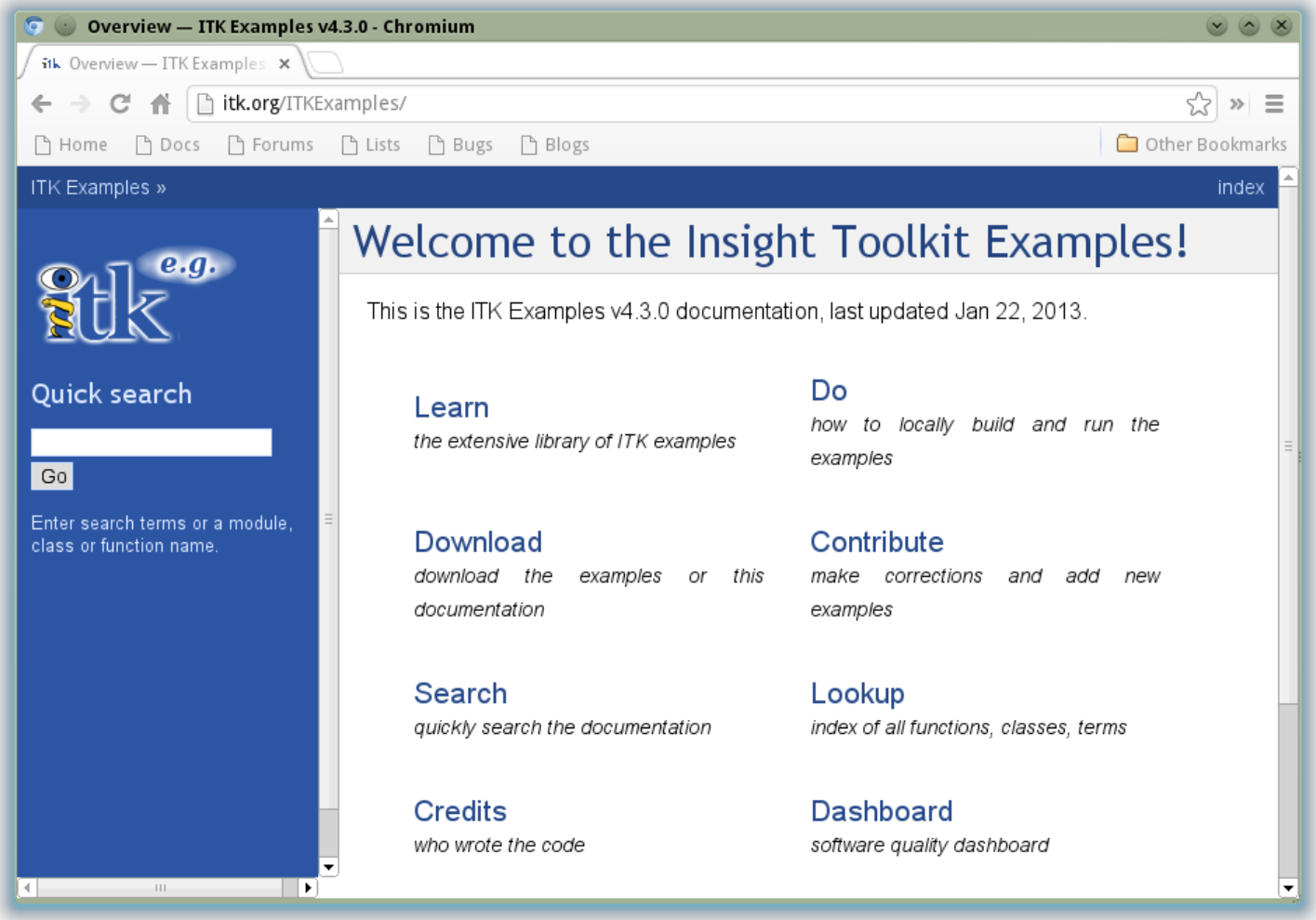
Edit the Doxygen Documentation in the Browser



Edit the Doxygen Documentation in the Browser



itk.org/ITKExamples



Modularization

Why Modularization

- Growth management

Why Modularization

- Growth management
- Software quality improvement

Why Modularization

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

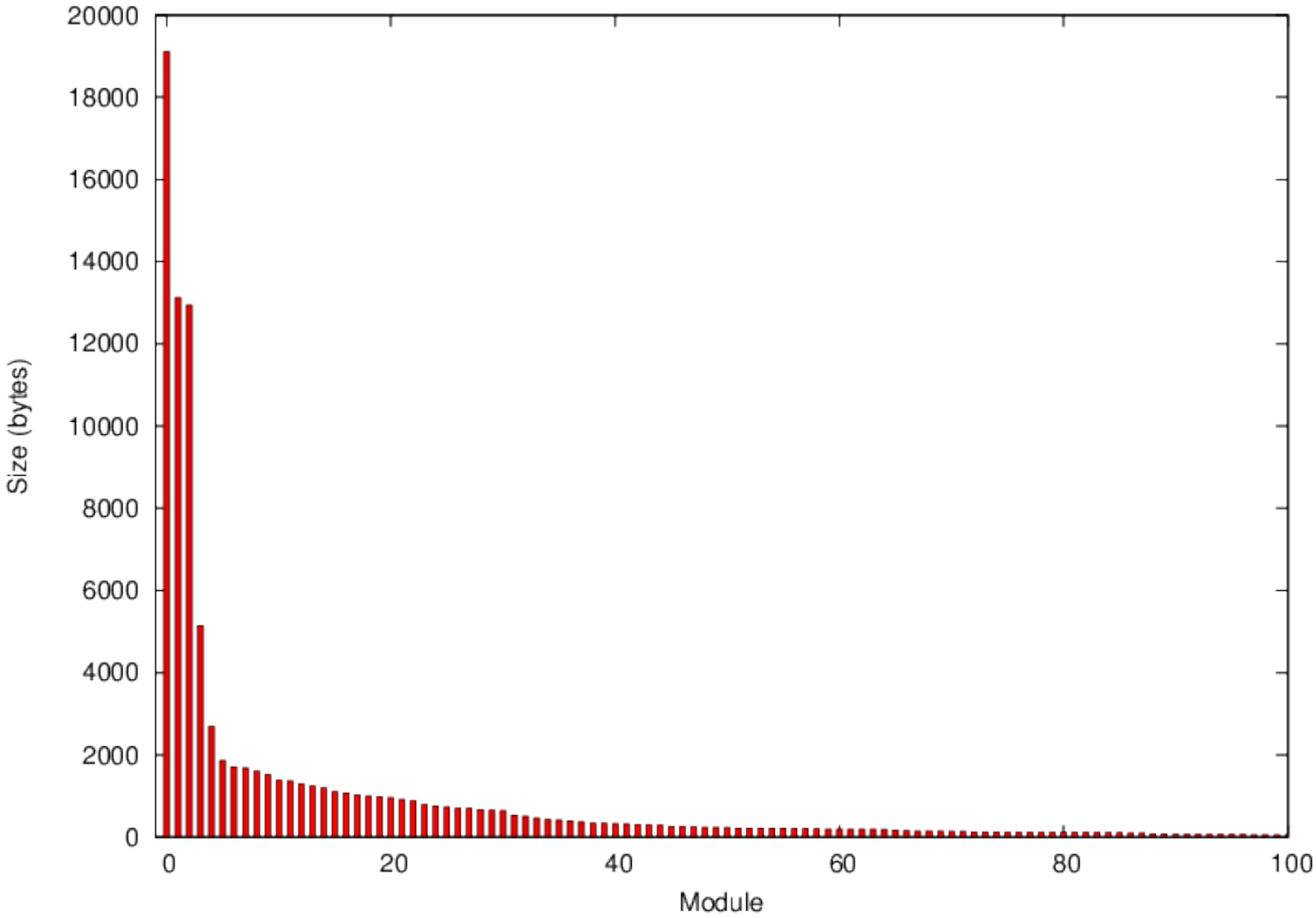
Why Modularization

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

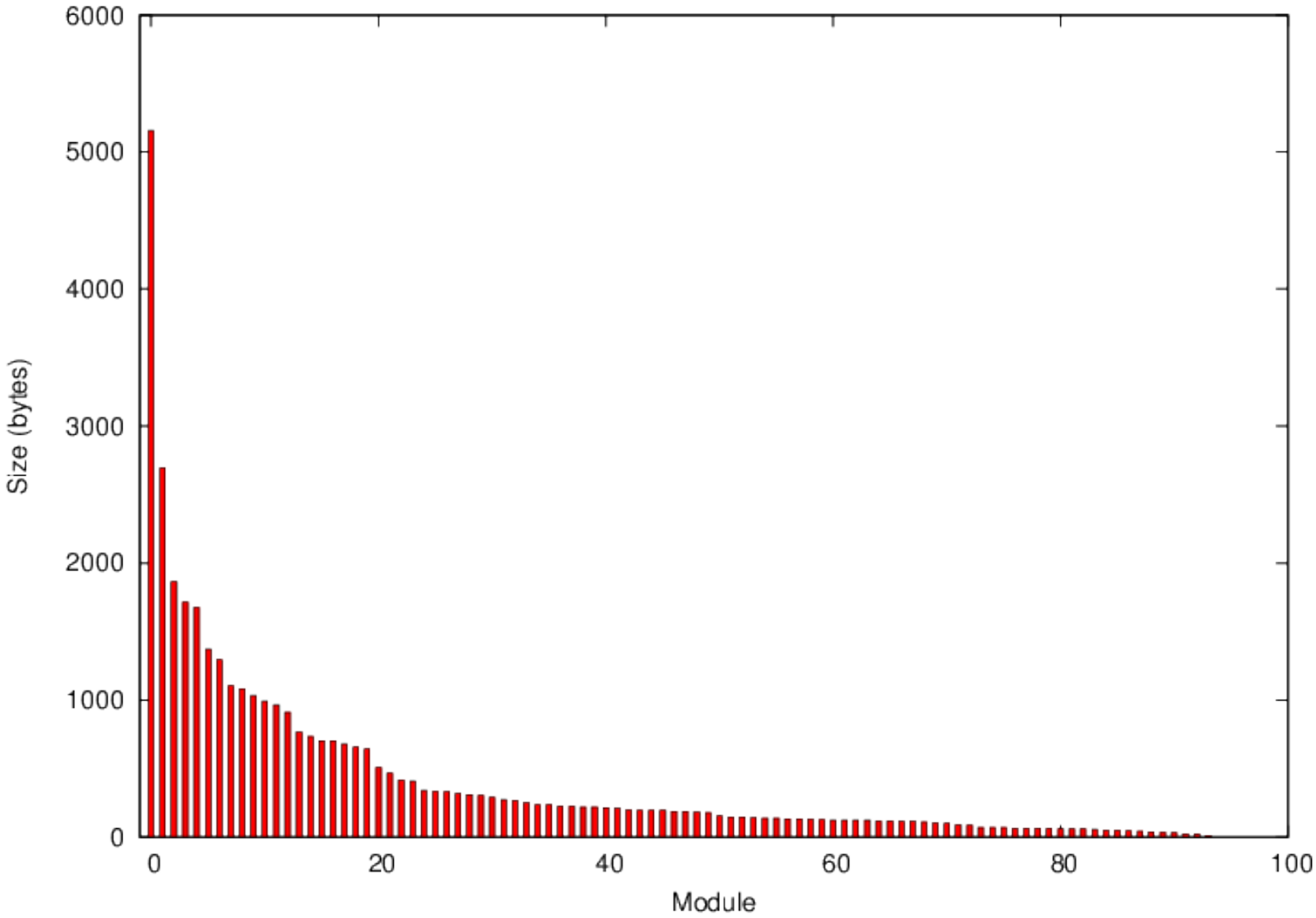
Why Modularization

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

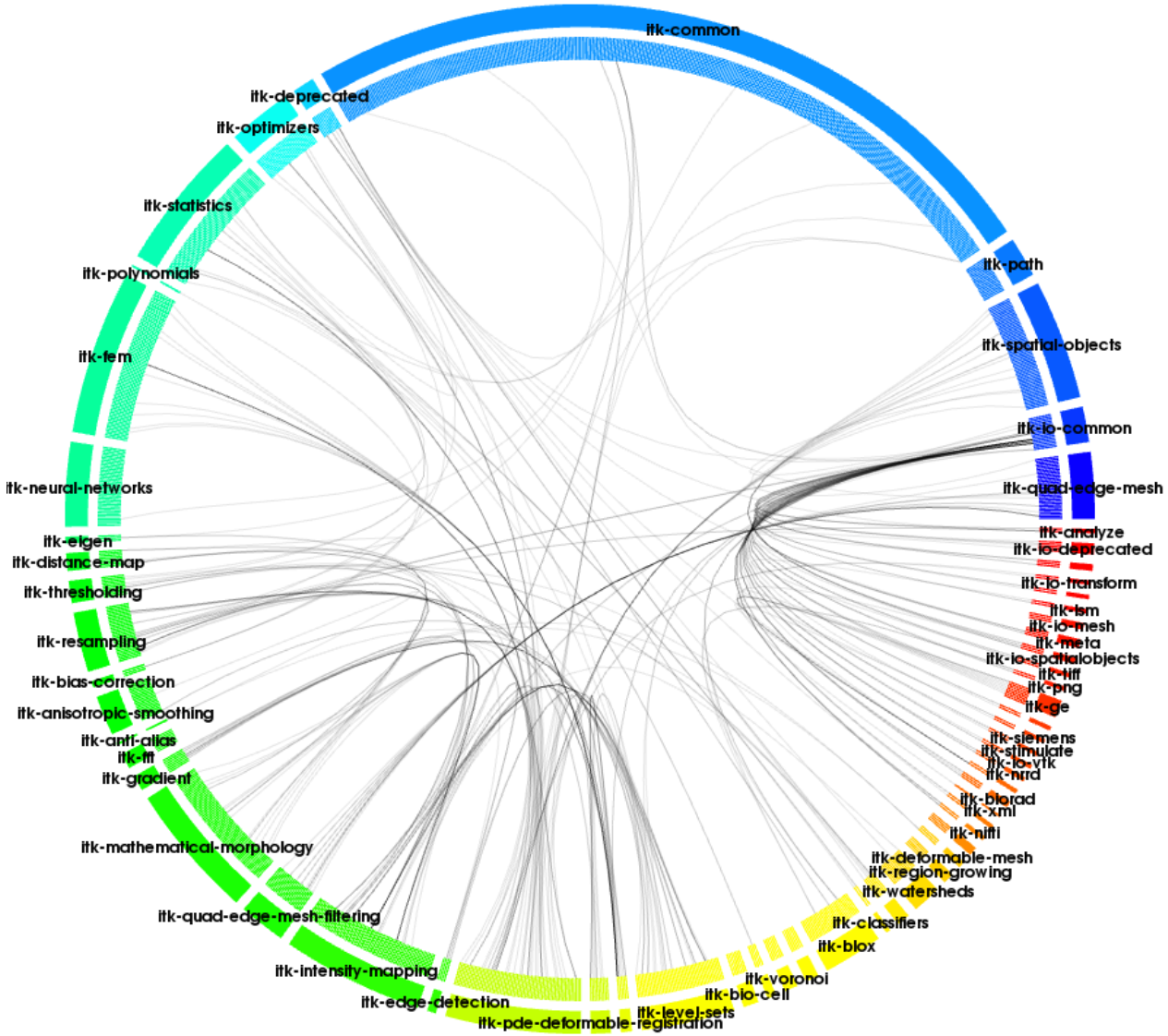
Module sizes



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.

Live Long
and
Prosper !