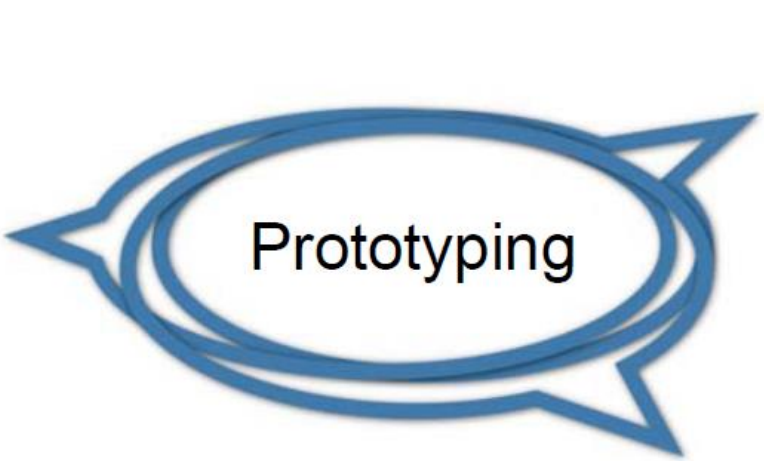
MeVisLab

MIP Prototyping

Innovation in clinical medical imaging requires close communication between…

Clinical users



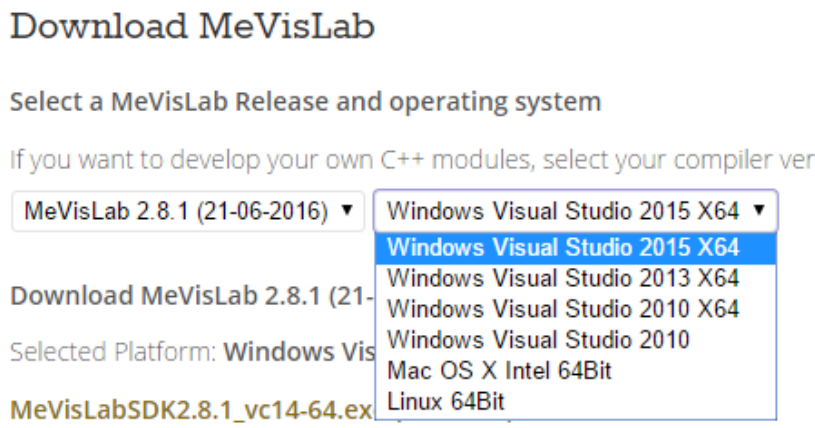
Researchers

Developers

Prototyping serves as a common language!

MeVisLab

- http://www.mevislab.de/
- In more than 20 years of development, MeVisLab has become one of the most powerful development platforms for medical image computing research.
- image processing, visualization and interaction modules can be combined to complex image processing networks using a graphical programming approach
- can easily be integrated using a modular, platform independent C++ class library.
- JavaScript or Python components can be added to implement dynamic functionality on both the network and the user interface level.
- based on the Qt application framework and the OpenInventor 3D visualization toolkit
- ITK and VTK AddOns



- Rapid Application Prototyping Environment
  - Cross-platform (Windows, Mac OS X, Linux)
  - Free for non-commercial usage
- Supported file formats
  - DICOM, TIFF, DICOM/TIFF, RAW, LUMISYS, PNM, Analyze, PNG, JPEG
- Currently 920+ Standard modules in the MeVisLab SDK core, 3000+ modules delivered in total
  - with 360+ ITK modules, 1470+ VTK modules, and 300+ modules in the Fraunhofer MEVIS release

MeVisLab development

- Three levels
  - **Visual level**
    - Programming with "plug and play"
      - Individual image processing, visualization and interaction modules can be combined to complex image processing networks using a graphical programming approach.
  - **Scripting level**
    - Creating macro modules and applications based on macro modules
      - Python scripting components can be added to implement dynamic functionality on both the network and the user interface level.
  - **C++ level**
    - Programming modules
      - New algorithms can easily be integrated using the modular, platform-independent C++ class library.

Image Processing

- Filters
  - Diffusion filters, morphology filters, kernel filters, Hessian, and vesselness filters
- Segmentation
  - Region growing, live wire, fuzzy connectedness, threshold, manual contours
- Transformations
  - Affine transformations, distance transformations, projection and Radon transforms, manual registration
- Statistics
  - Histograms, global image statistics, box counting dimension
- Other
  - Unary/binary arithmetic, resampling/reformatting, dynamic data analysis, noise/test pattern generators

Modules for Visualization

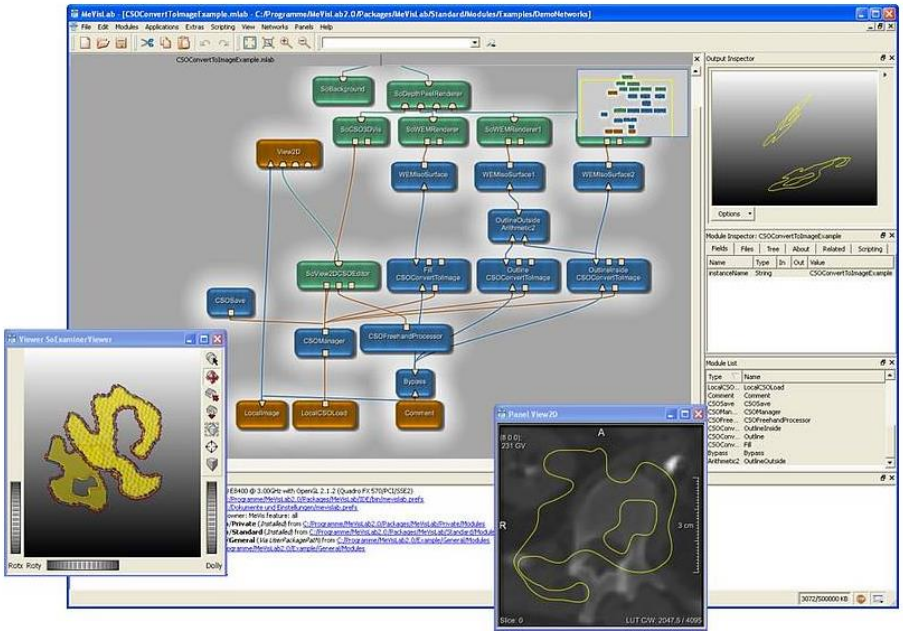
- MeVisLab provides modules for visualizing image data and other data objects in 2D and 3D.
- A set of lookup table (LUT) modules allows applying basic window/level adjustment or flexible color encoding schemes.
- The visualization functionality in MeVisLab is based on the well-established visualization and interaction library **Open Inventor**.

High-quality Volume Renderer: MeVisLab Giga Voxel Renderer

- MeVisLab features a high-quality volume renderer that is based on OpenGL and its extensions.
- It supports the rendering of large volume datasets, even if they do not fit into the main memory.
- An optimized, multi-resolution technique based on an octree representation and 3D textures adaptively selects the best resolution depending on camera position, volume of interest, and available resources.

MeVisLab Software Development Kit (SDK)

- Using the **MeVisLab Software Development Kit (SDK)**, a developer is able to implement and test own algorithms, visualization or interaction methods, or even complete processing workflows.
- The MeVisLab SDK offers a variety of features that support module programming, scripting, and network development.

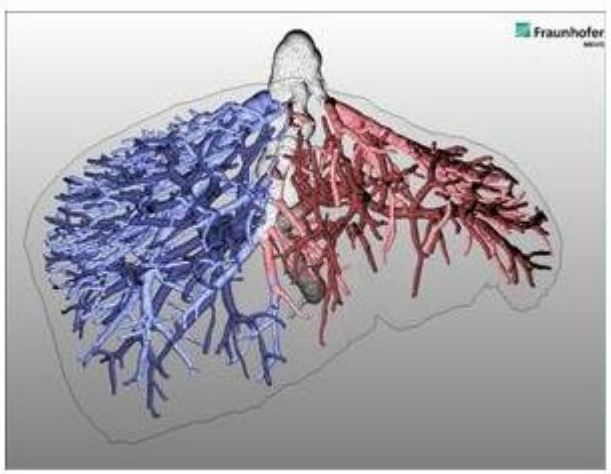


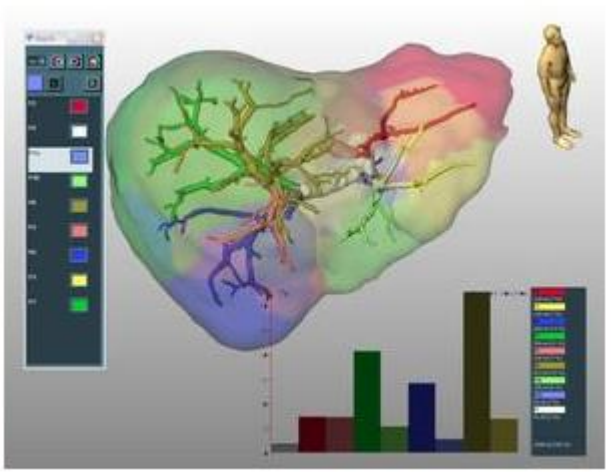
Open Inventor

- An object-oriented 3D toolkit developed by Silicon Graphics (SGI)
  - offering a comprehensive solution to interactive graphics programming problems
- Most of the visualization modules of MeVisLab make use of Open Inventor.

Open Inventor (OIV)

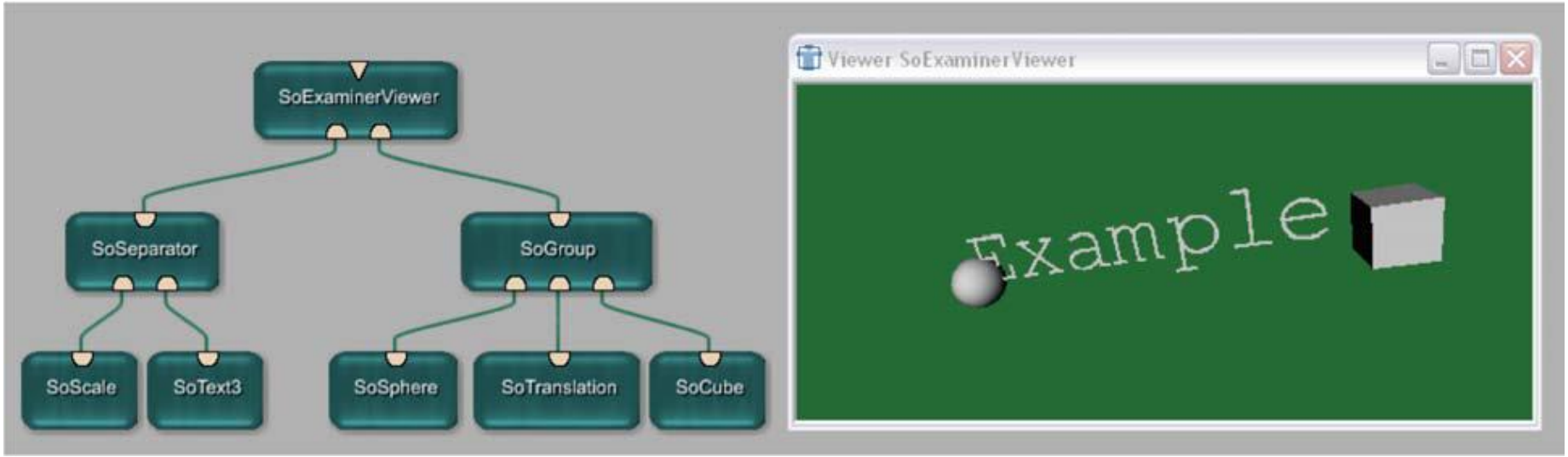
- Direct Open Inventor node support
- Open Inventor:
  - Scene graph paradigm
  - Object, rendering, transformation, property, … nodes
  - Based on OpenGL
  - Extensions to support 2D image viewing/manipulation
- Mixed ML/Open Inventor Modules
- http://www.mevislab.de/mevislab/features/open-inventor/





Open Inventor Scene Graph

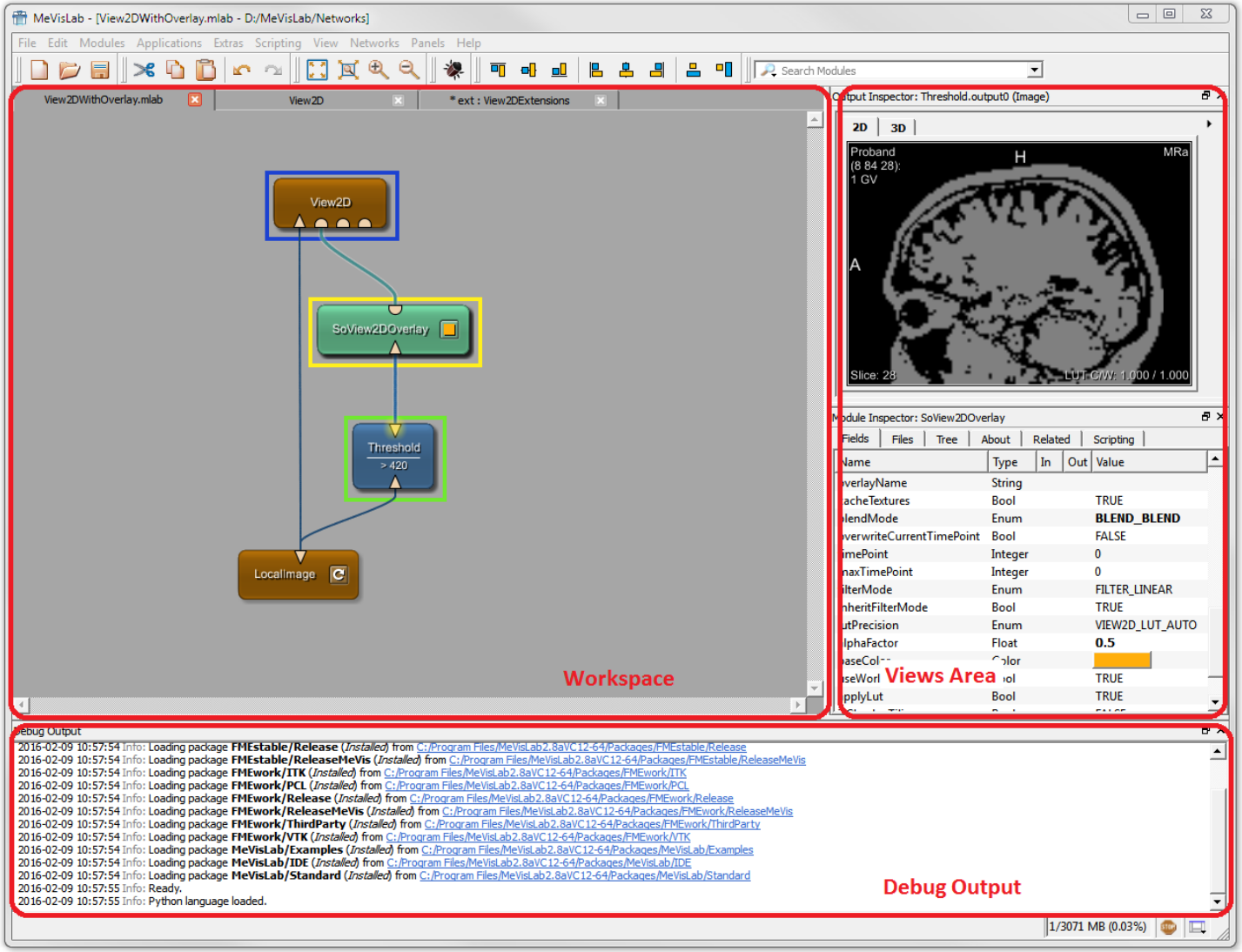
- Scene objects are represented by nodes
- Size and position is defined by transformation nodes
- A rendering node represents the root of the scene graph



Integration of Visualization, Segmentation and Registration Toolkits

- The Insight Segmentation and Registration Toolkit (ITK) is an extensive collection of leading-edge algorithms for registration, segmentation, and analysis of multidimensional data.
  - It is an open-source, cross-platform software package written in C++ and supported by the US National Library of Medicine.
- The Visualization ToolKit (VTK) is an open source, freely available software library for 3D computer graphics, image processing, and visualization.
  - It has become one of the most popular open source toolkits for visualization purposes and is used by thousands of researchers and developers around the world.

MeVisLab User Interface



MeVisLab Modules

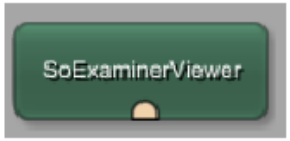
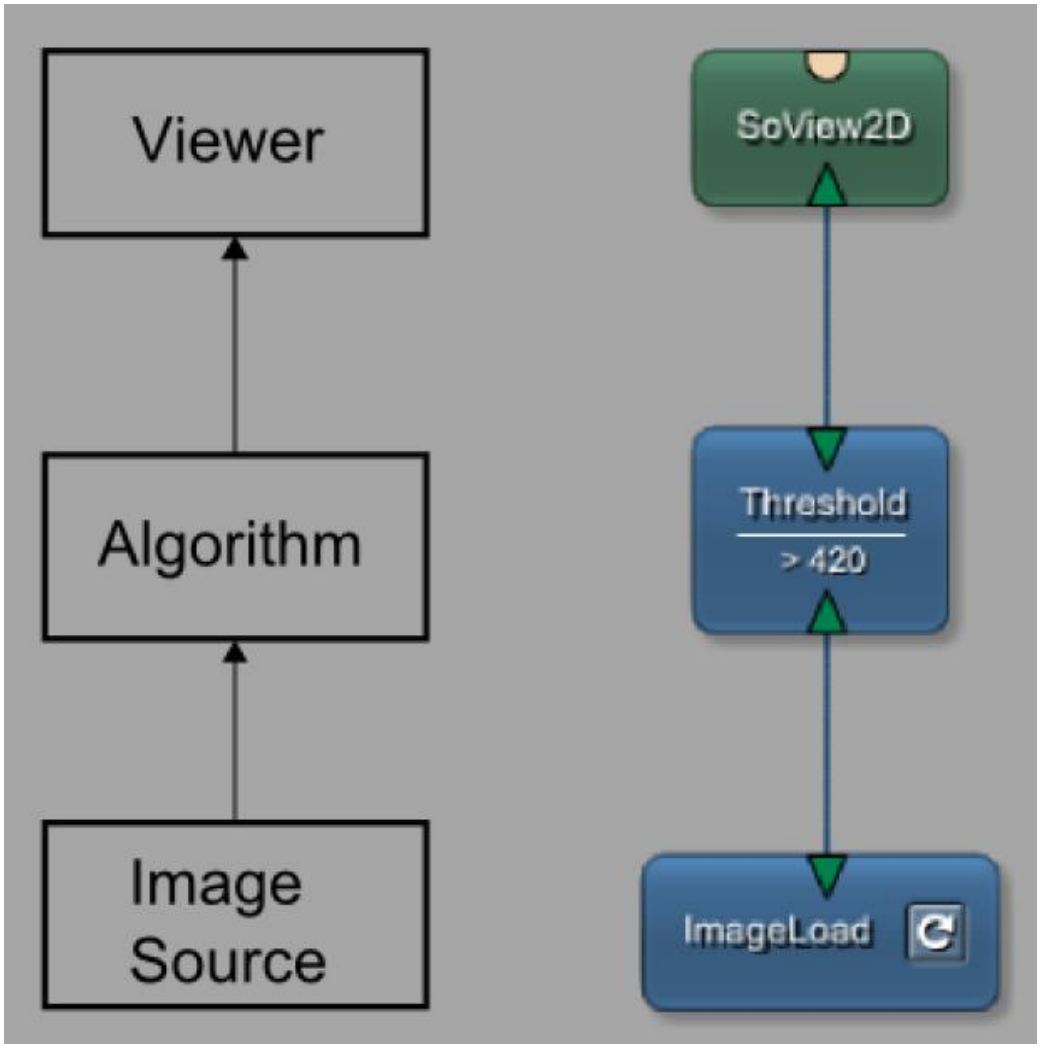
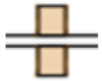
Type	Look	Characteristics
ML Module (blue)		page-based, demand-driven processing of voxels
Open Inventor (So) Modules (green)		visual scene graphs (3D)
Macro Module (brown)		combination of other module types, allowing implementing hierarchies and scripted interaction

Image Processing Pipeline



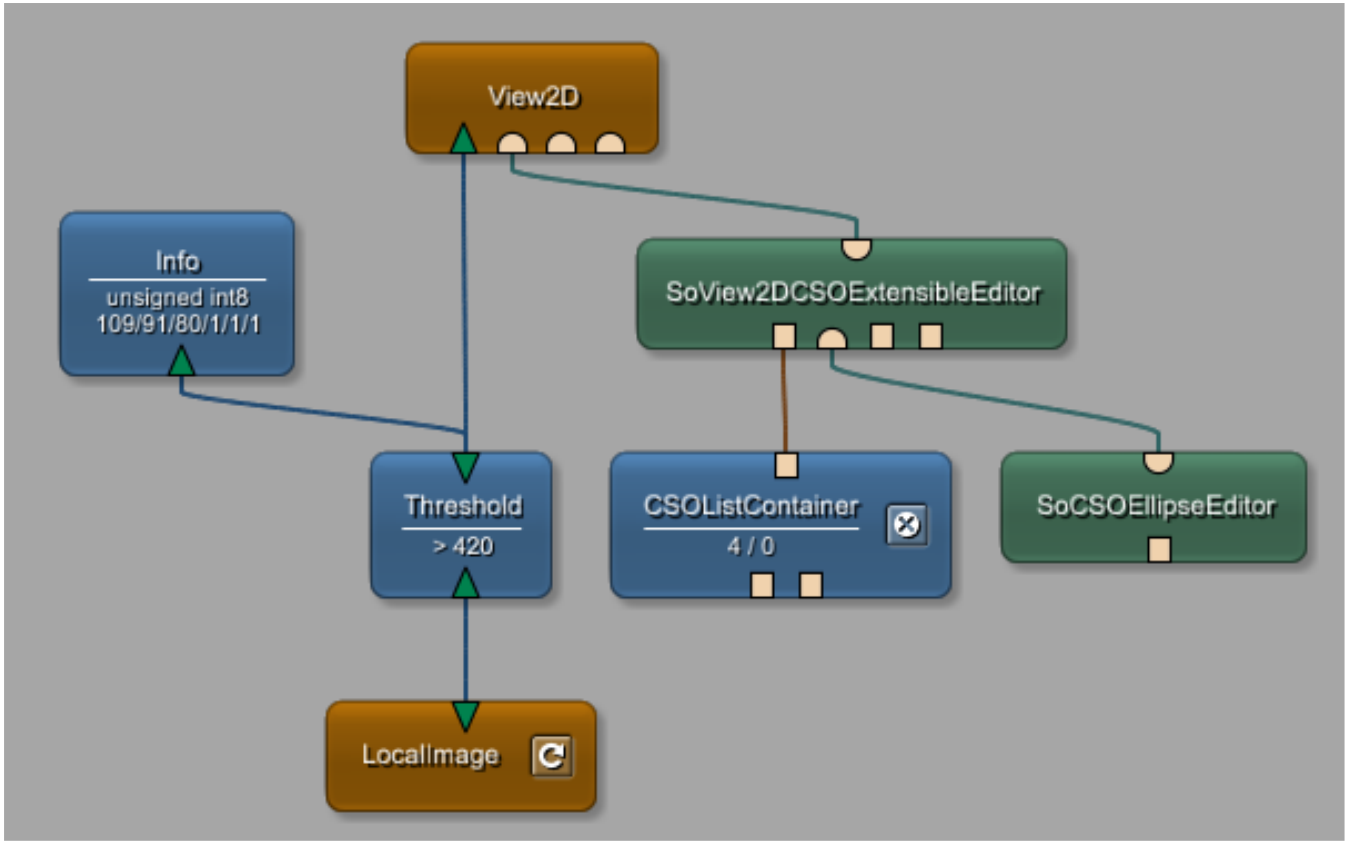
Connectors

Look	Shape	Definition
	triangle	ML images
	half-circle	Inventor scene
	square	Base objects: pointers to data structures

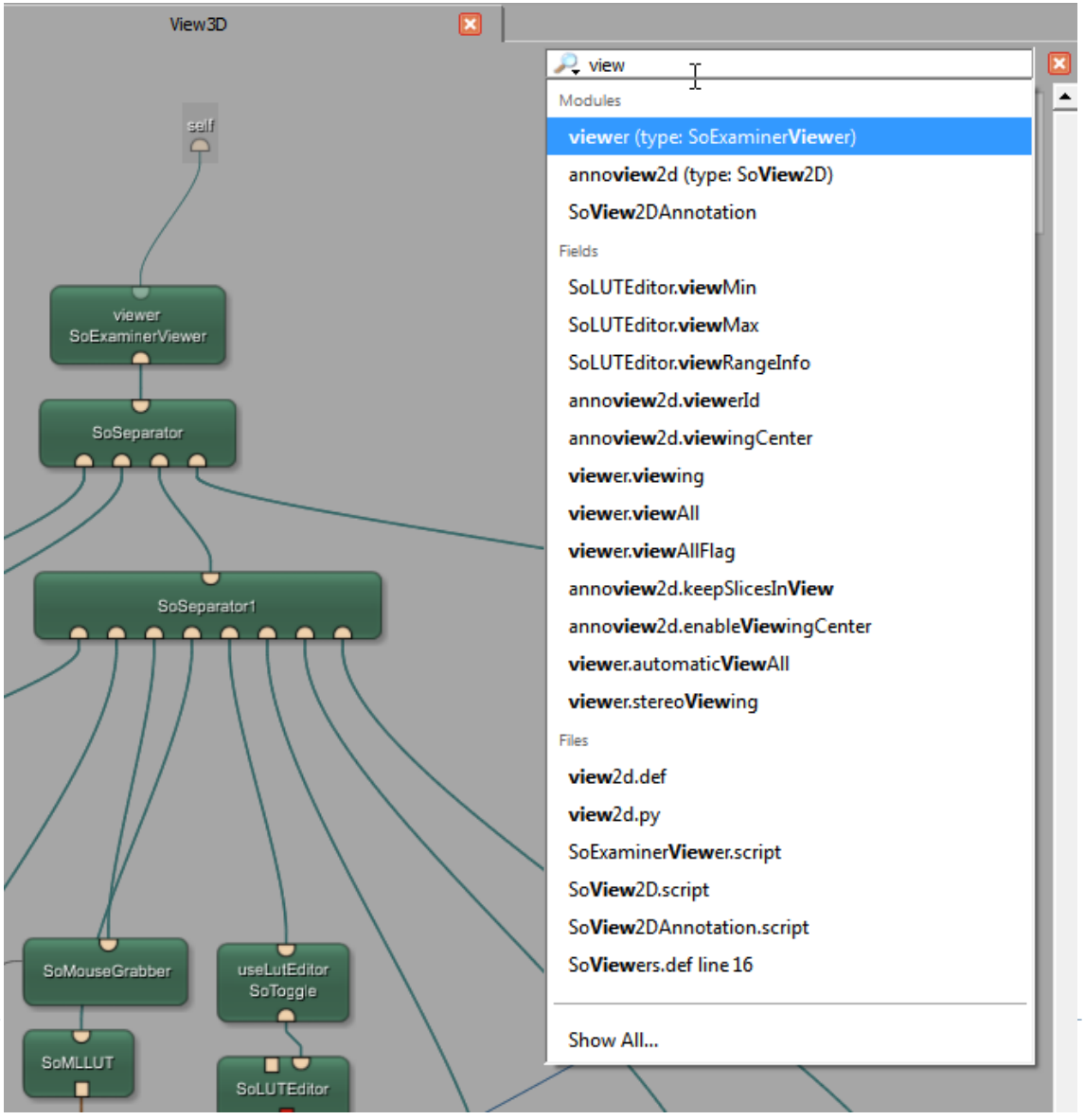
Connections

Type	Look	Characteristics
Data connections (connector connections)		The direct connection between connectors. Depending on which connectors are involved, the connection is rendered in a different color: blue for ML, green for Open Inventor, brown for Base.
Parameter connections (field connections)		Connections created by connecting parameter fields within or between modules

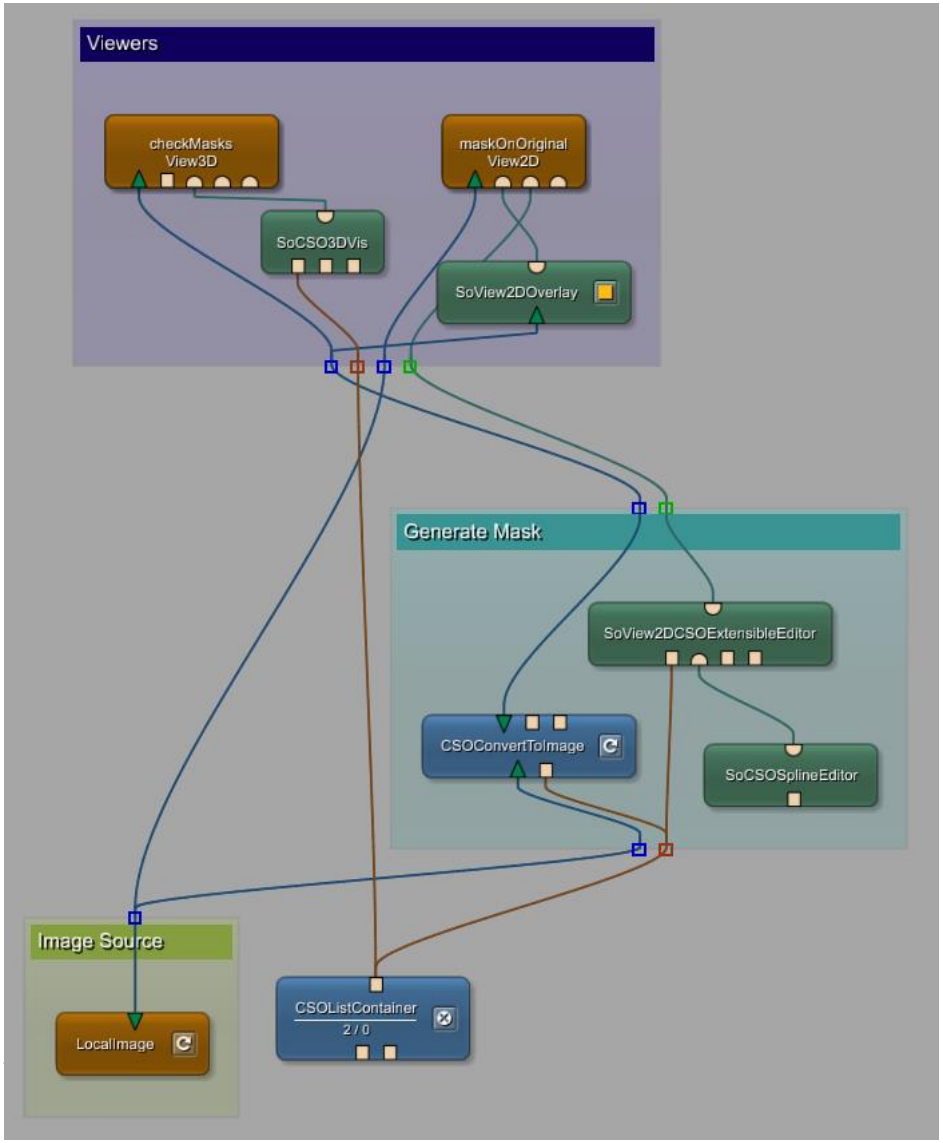
Network Layout

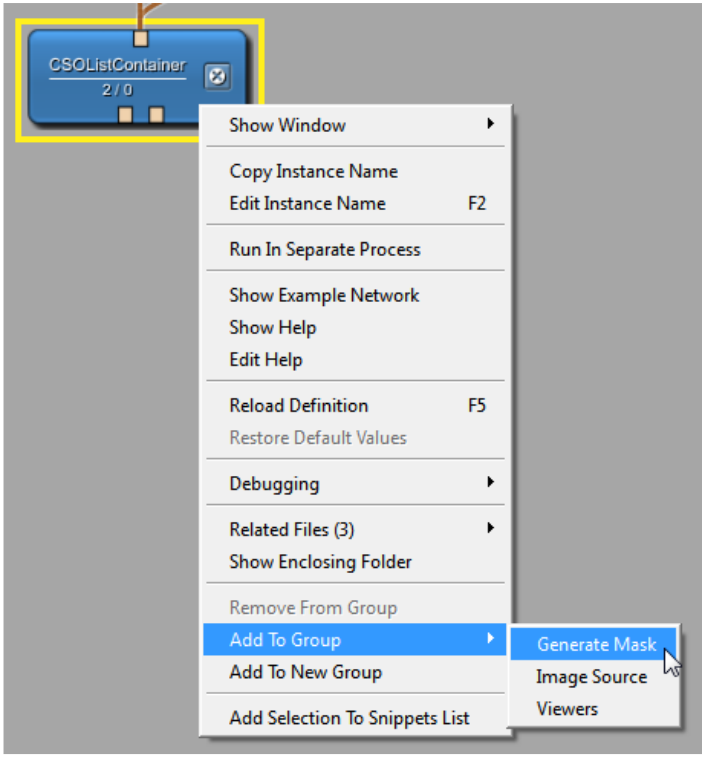


Network Quick Search



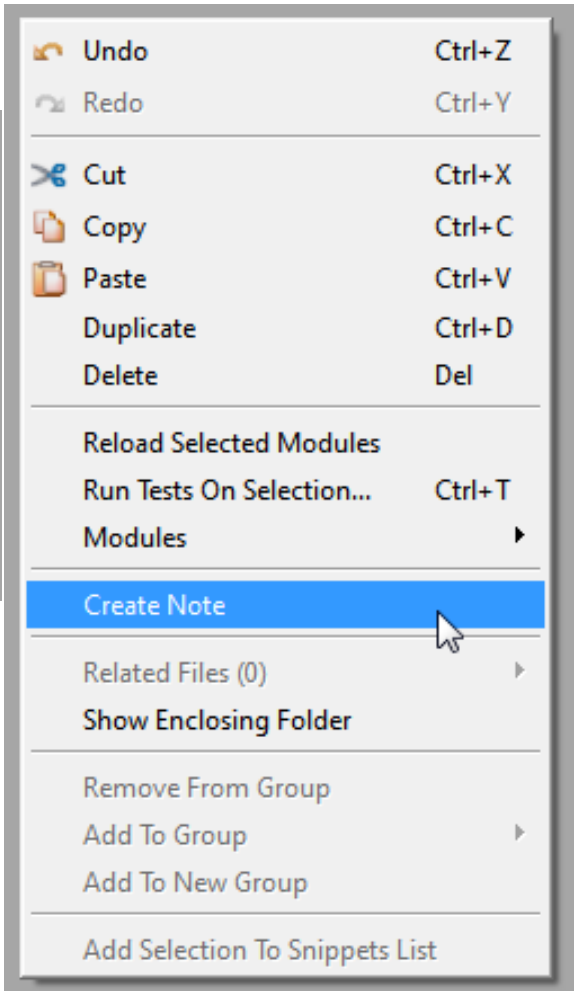
Using Groups





Using Notes

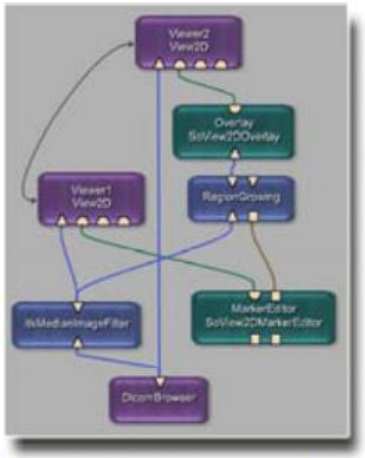

CSOListContainer
2 / 0
CSOListContainer
The module CSOListContainer is a light-weight container for holding a CSOList.
The module is similar to CSOManager but without all the fields to control the default and current values of CSOs and CSOGroup.
The CSOListContainer also has no list views
...


Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Duplicate Ctrl+D
Delete Del
Reload Selected Modules
Run Tests On Selection... Ctrl+T
Modules
Create Note
Related Files (0)
Show Enclosing Folder
Remove From Group
Add To Group
Add To New Group
Add Selection To Snippets List

Scripting (MDL)

- User interfaces are created with the "Module Definition Language" (MDL)
  - Abstract hierarchical GUI language
  - Interpreted at run-time, allows rapid prototyping
- www.mevislab.de/fileadmin/docs/html/mdl/

Getting Started: Chapter 11. GUI Design in MeVisLab



Module Network

```
Window "TestApplication" {
  Vertical { expandX=yes expandY=yes
    Horizontal { expandX=yes expandY=yes
      Viewer Viewer1.self { type=SoRenderArea }
      Viewer Viewer2.self { type=SoRenderArea }
    }
    Box "ITK Filter Parameter" {
      Field itkMedianImageFilter.radius {
        title = "Radius:"
      }
    } // Box
    Box "General Region Growing Parameters" {
      Field RegionGrowing.basicNeighborhoodType {
        title = "Neighborhood Relation:"
      }
      CheckBox RegionGrowing.autoThreshold {
        title = "Auto-Generate Thresholds"
      }
    } // Box
    Box "Region Growing" { layout=Horizontal
      Button RegionGrowing.update { title="Update" }
      ProgressBar = RegionGrowing.theProgressBar
      Button RegionGrowing.clear { title="Clear" }
    } // Box
    Box "Dicom Browser" { expandY=no
      Panel { module=DicomBrowser panel=browserParams }
      Panel { module=DicomBrowser panel=browserPanel }
    } // Box
  }
}
```

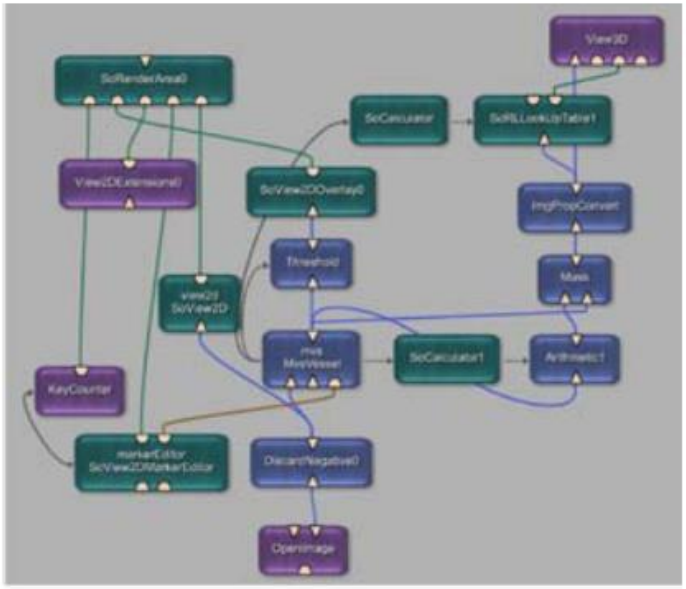
MDL Script

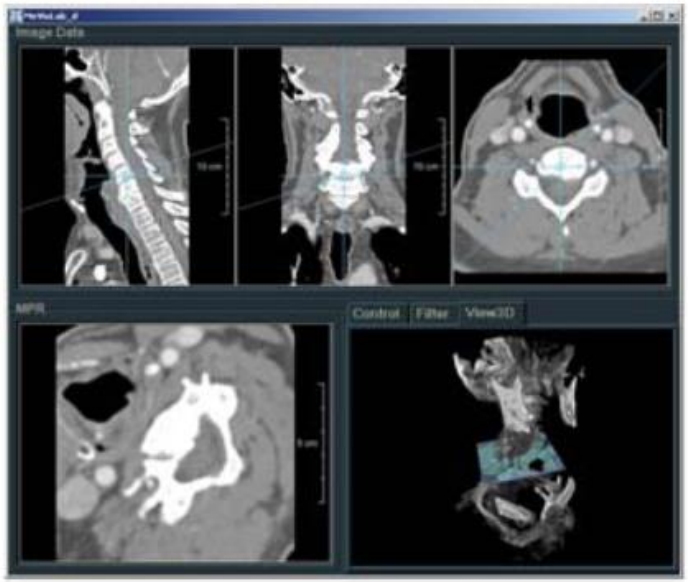
Graphical User Interface

Schematic Representation

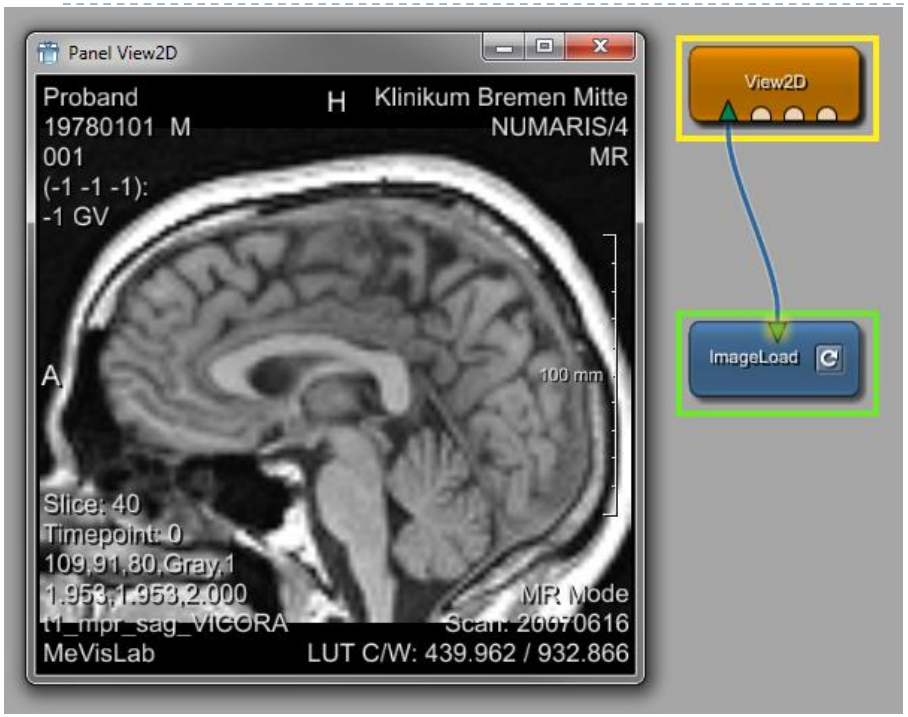
Application Prototyping

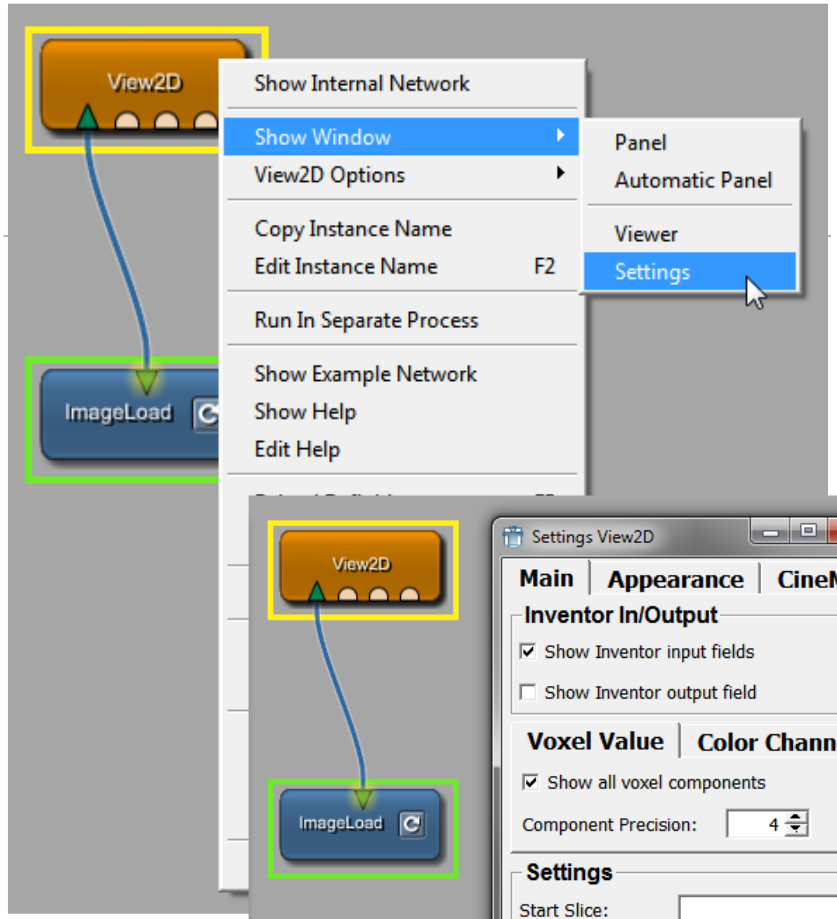
- Hide network complexity
- Design user interfaces
- Scripting for dynamic components

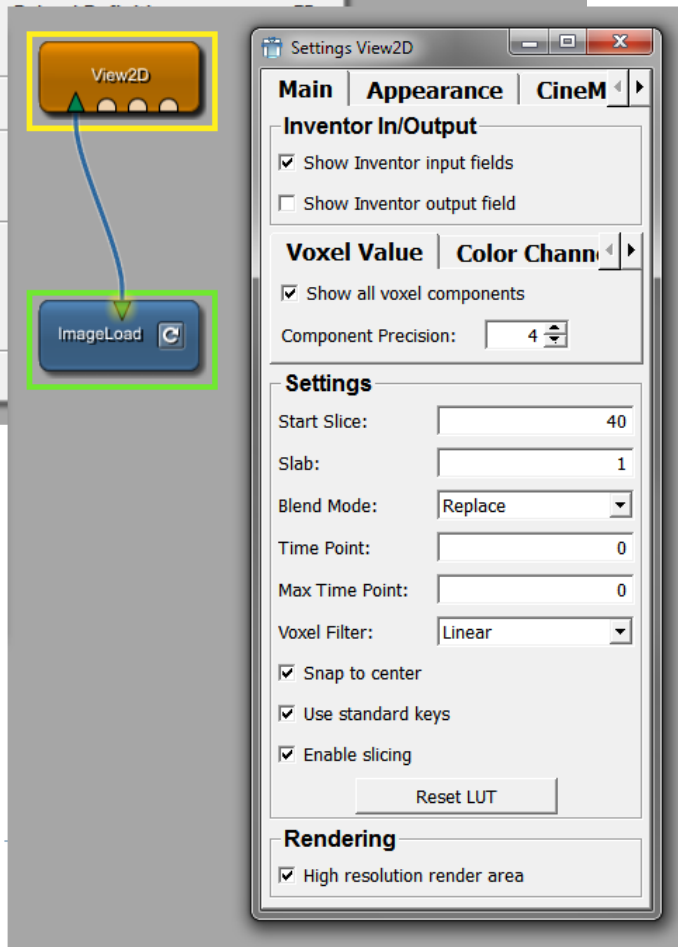




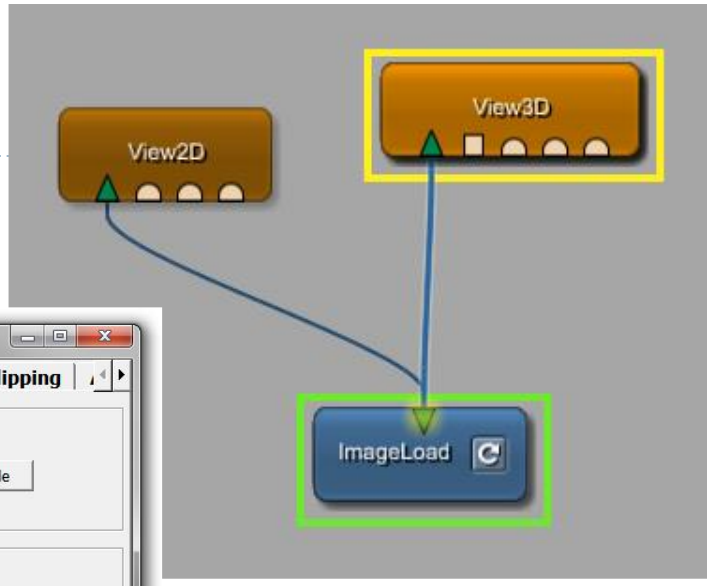
View2D Module

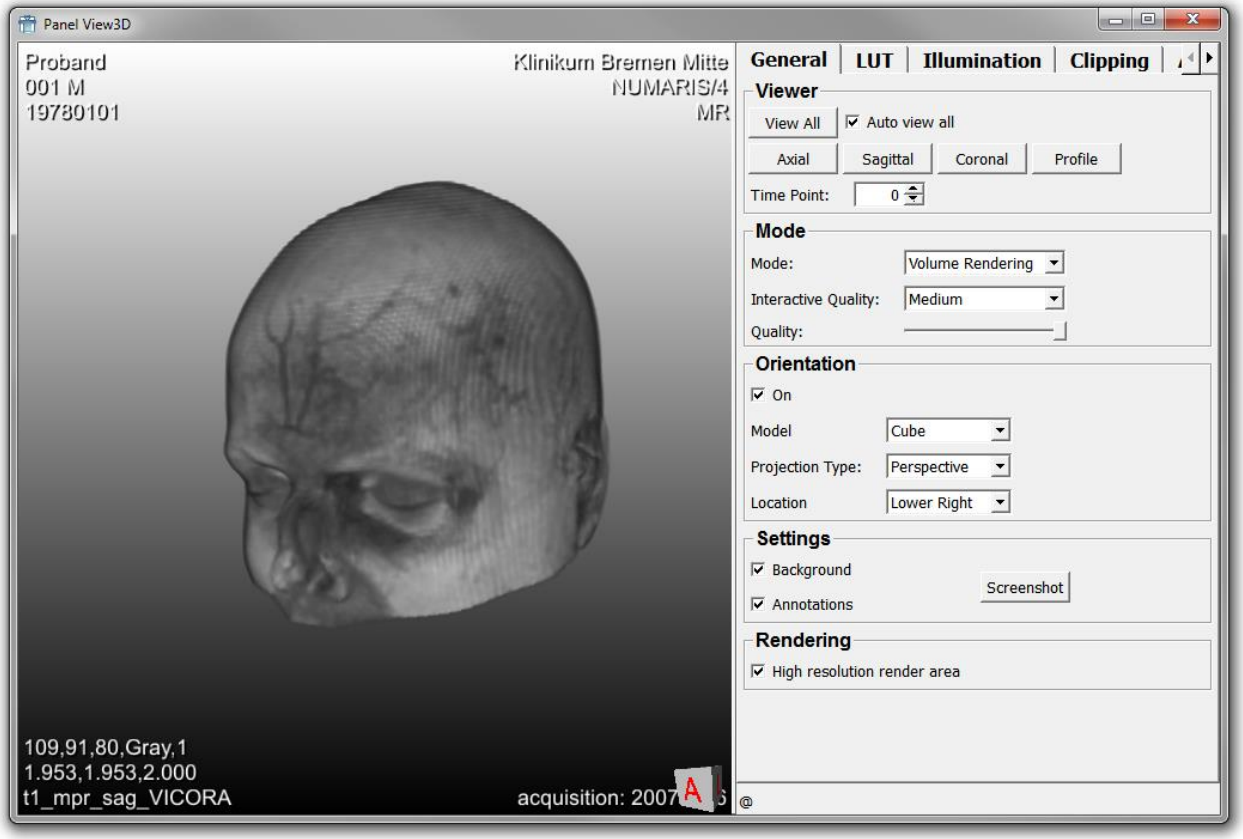




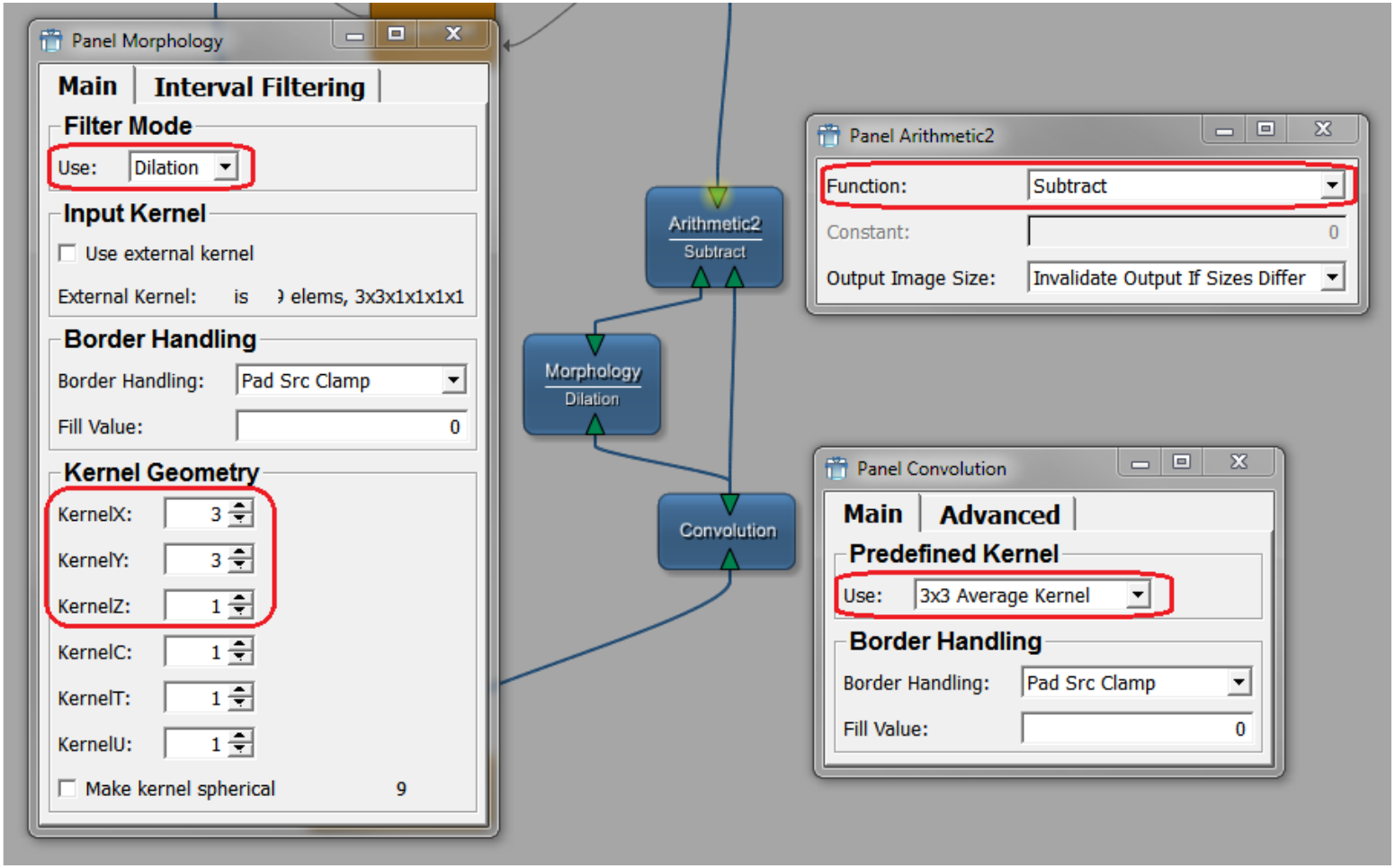


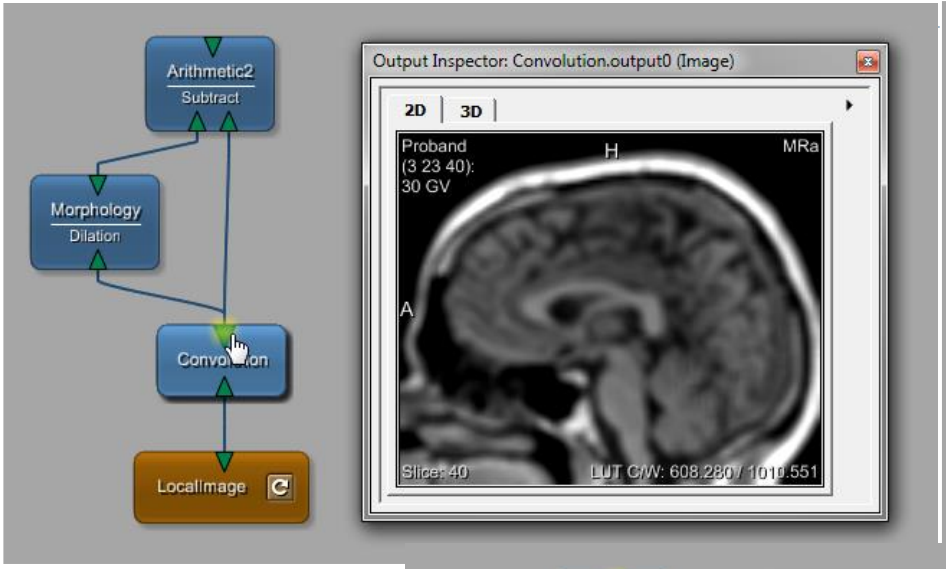
View3D Module

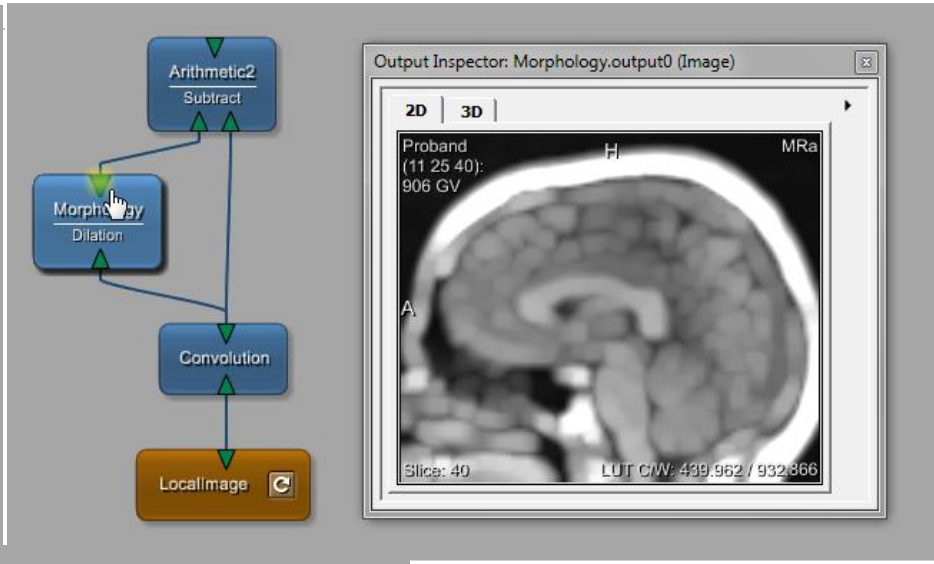


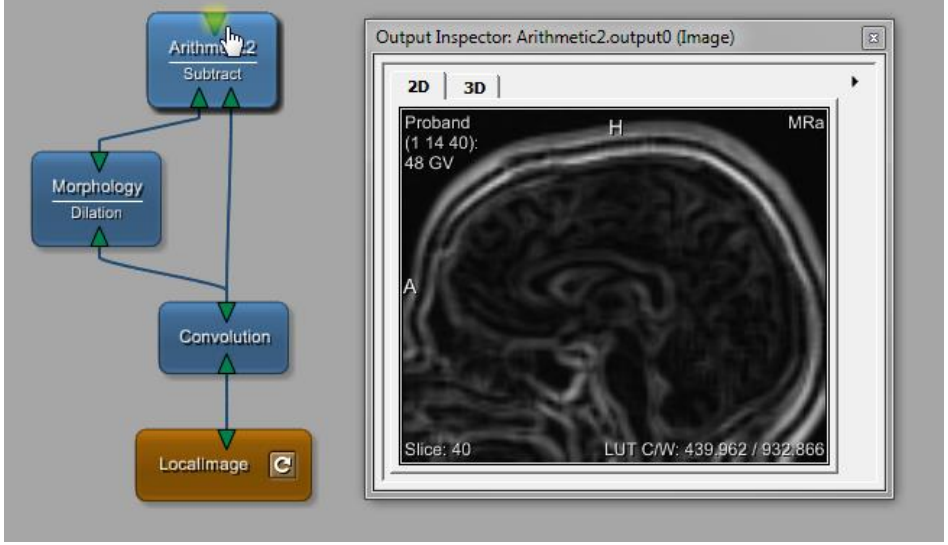


Implementing the Contour Filter

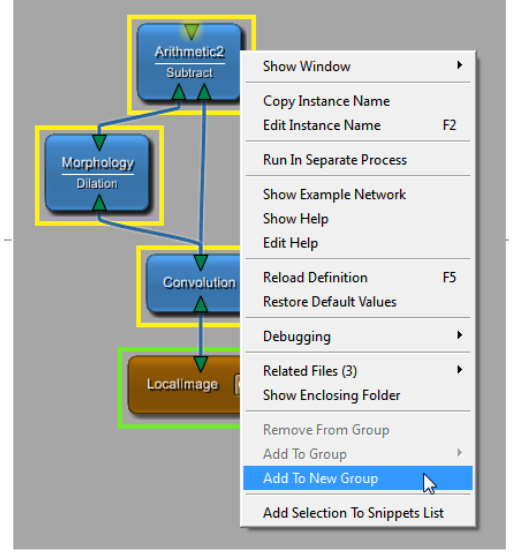


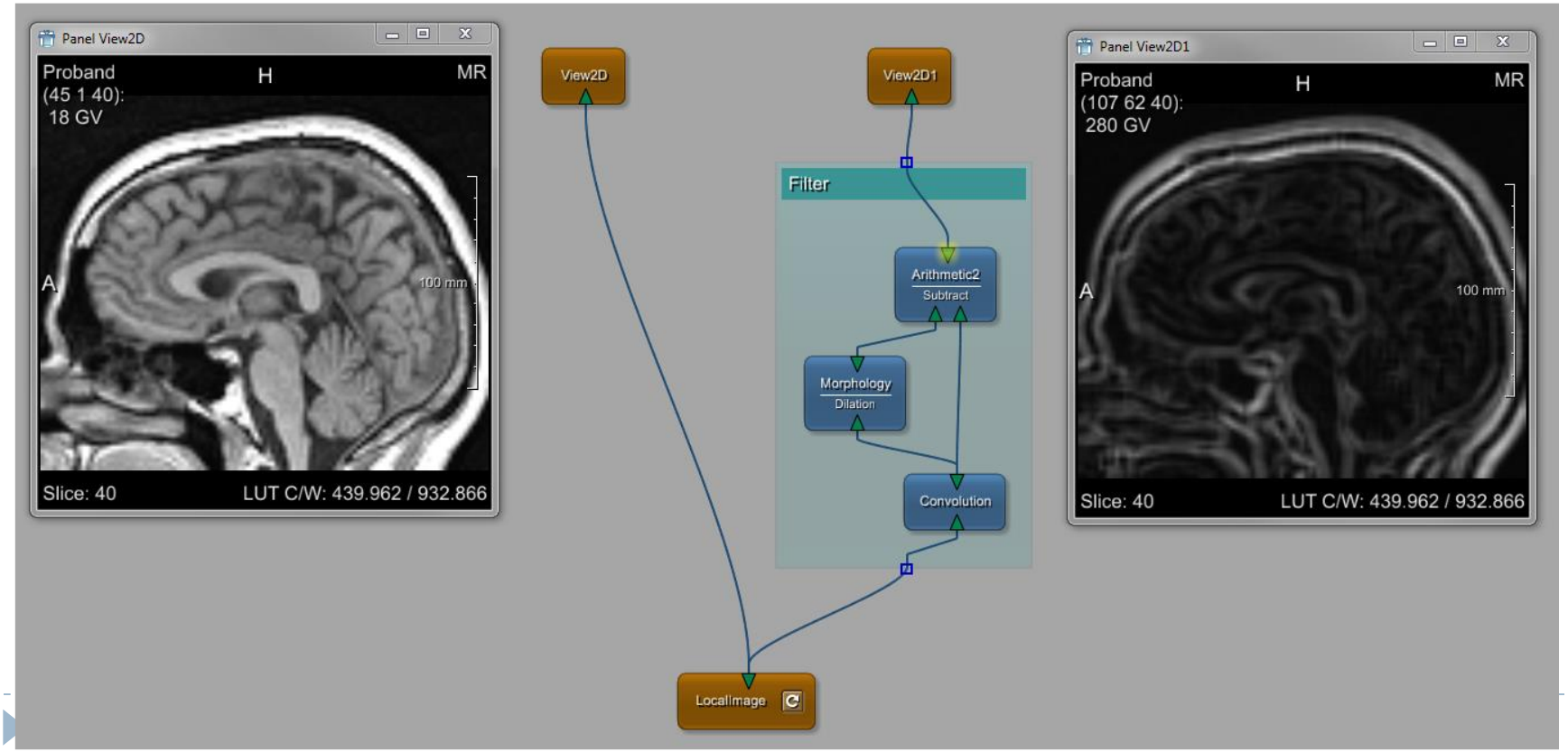

Arithmetic2
Subtract
Morphology
Dilation
Convolution
LocalImage
Output Inspector: Convolution.output0 (Image)
2D
3D
Proband
(3 23 40):
30 GV
H
MRa
A
Slice: 40
LUT C/W: 608.280 / 1010.551


Arithmetic2
Subtract
Dilation
Convolution
LocalImage
Output Inspector: Morphology.output0 (Image)
2D
3D
Proband
(11 25 40):
906 GV
H
MRa
A
Slice: 40
LUT C/W: 439.962 / 932.866

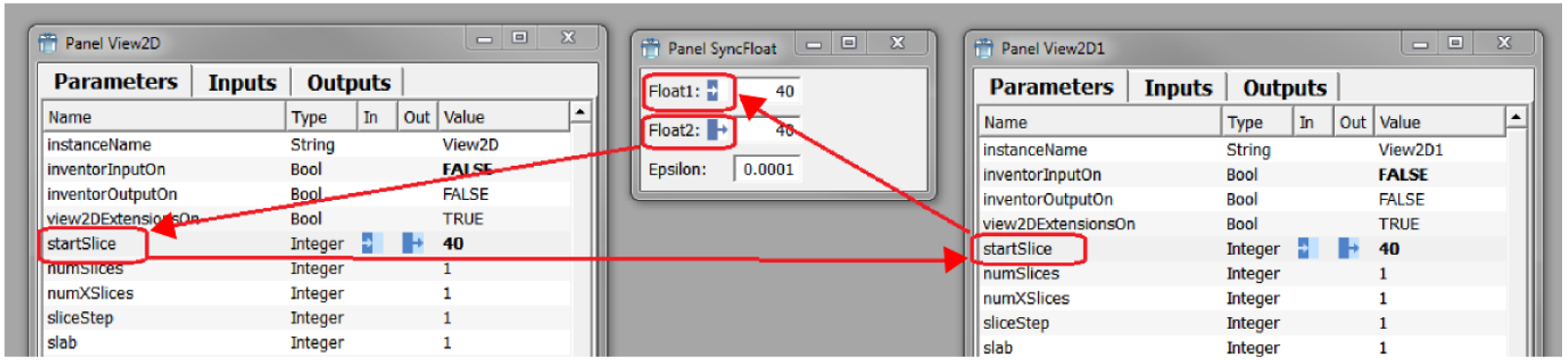

Subtract
Morphology
Dilation
Convolution
LocalImage
Output Inspector: Arithmetic2.output0 (Image)
2D
3D
Proband
(1 14 40):
48 GV
H
MRa
A
Slice: 40
LUT C/W: 439.962 / 932.866

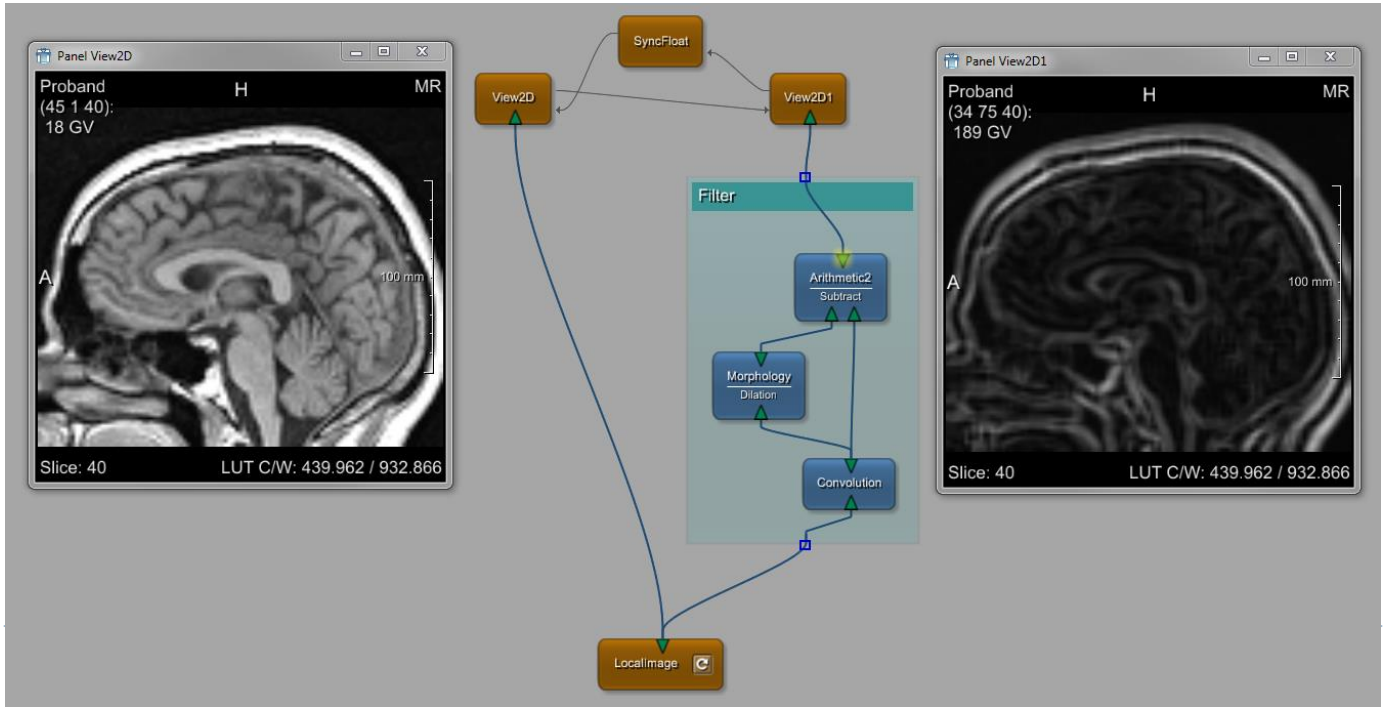
Creating a New Group





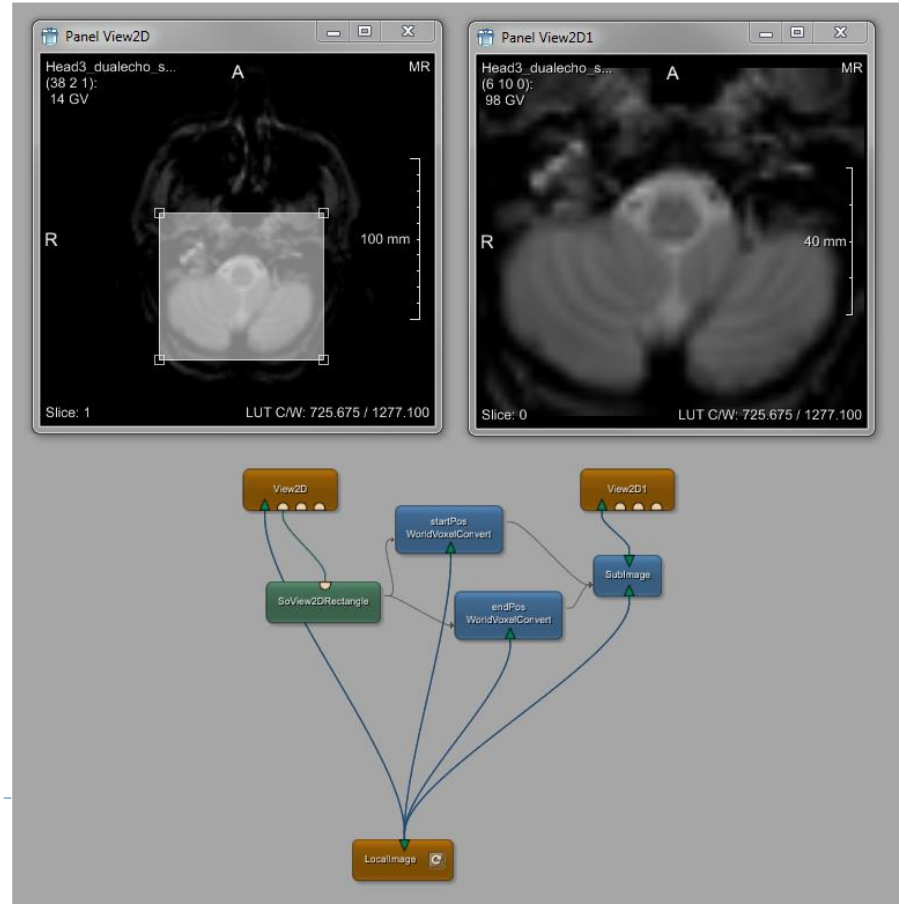
Parameter Connection for Synchronization





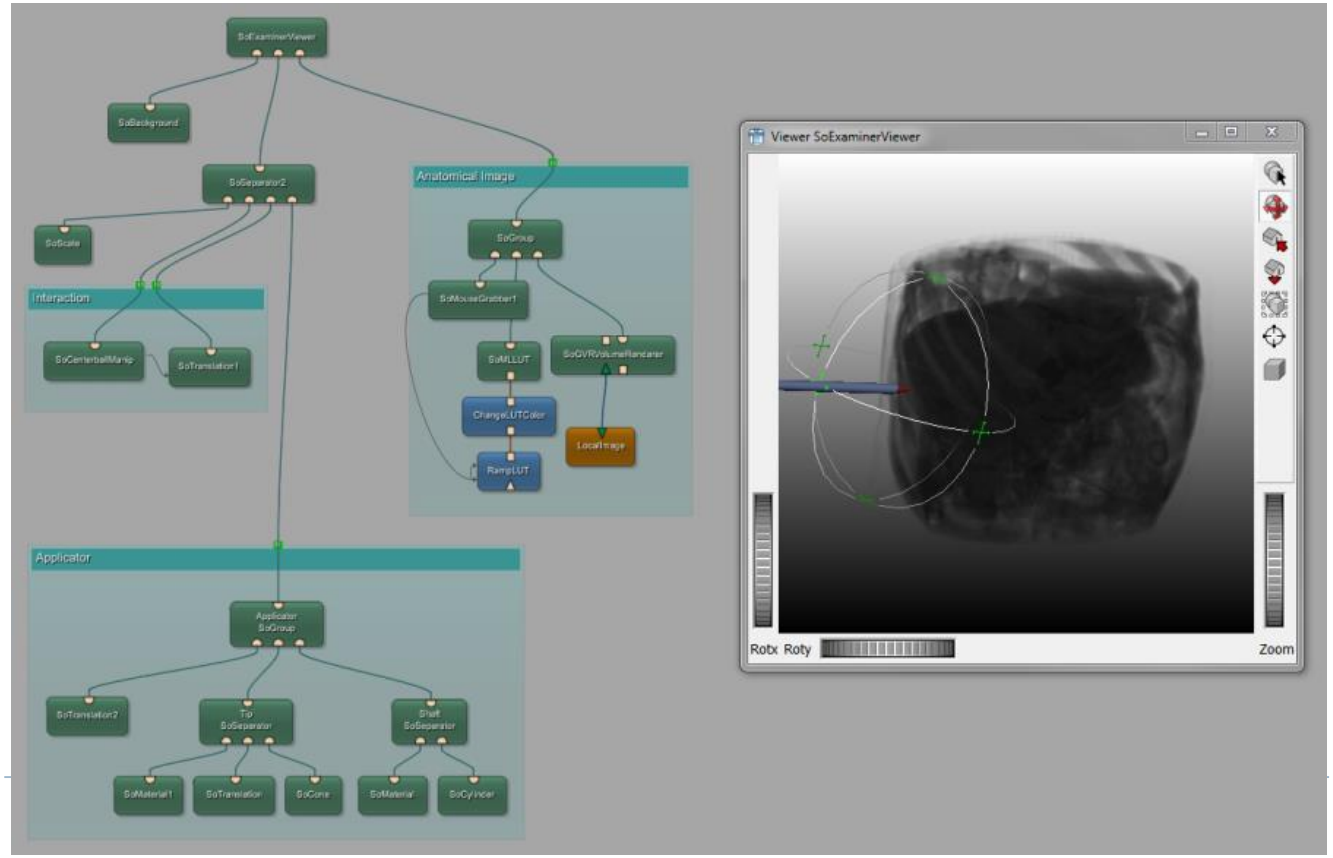
Getting Started: Chapter 5. Defining a Region of Interest

- a network that allows defining a 2D region of interest (ROI), that is by selecting a region of the image in the first viewer, the selected region is displayed as a subimage in a second viewer



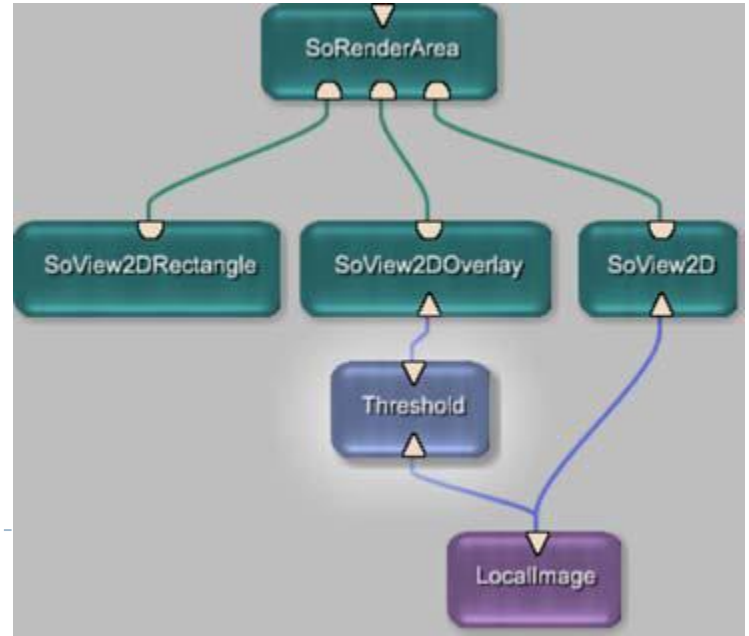
Getting Started: Chapter 7. Creating an Open Inventor Scene

- a dynamically definable applicator (needle for minimally invasive surgeries) shall be placed at a position and an angle relative to the rendering of an anatomical image



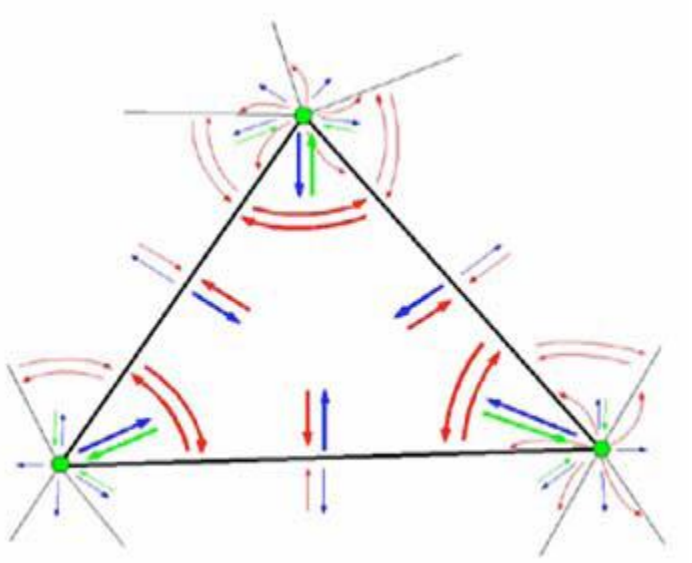
2D Viewers

- Modular 2D Viewer Library (SoView2D)
  - Hardware accelerated using textures and shaders
  - Supports interactive LUT even on large images
- Extension mechanism supports:
  - Overlays
  - Markers
  - ROIs
  - Contours
  - User extensions can add drawing and event handling



Winged Edge Mesh Library (WEM)

- Data structure proposed by Baumgart, 1975
- Mesh consists of Nodes, Edges and Faces
- Dense pointer structure of incident primitives
- Fast access to neighboring structures



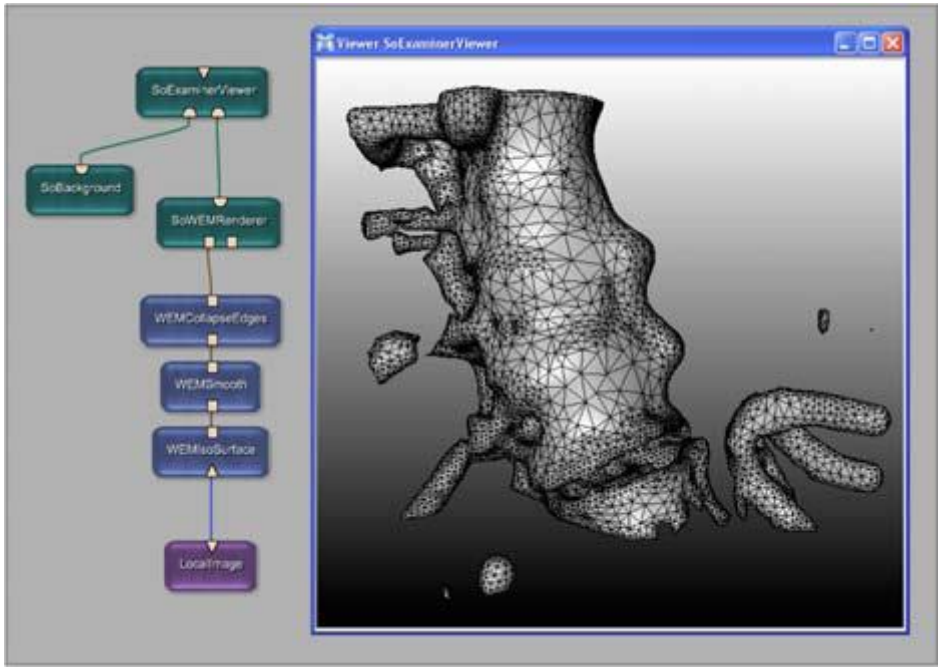
Pointer links in a neighborhood

WEM Modules Overview

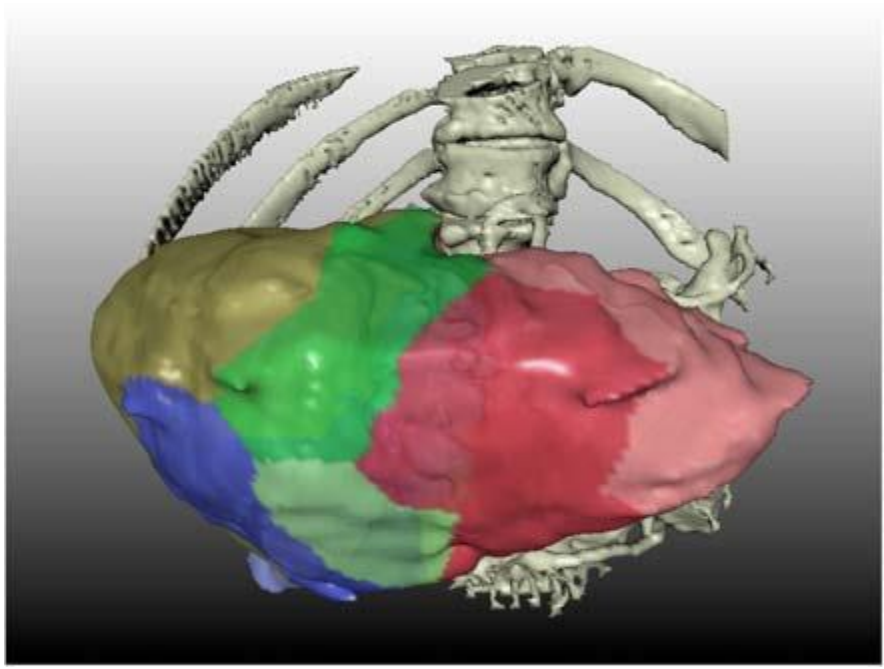
- Generation:
  - WEMIsoSurface
- Processing:
  - WEMCollapseEdges
  - WEMSmooth
  - WEMPurge
  - WEMClip
  - …
- Rendering:
  - SoWEMRenderer
    - Different Render Modes
    - Optional Coloring by LUT Values

WEM Sceneshots

Network with iso surface generation and polygon reduction

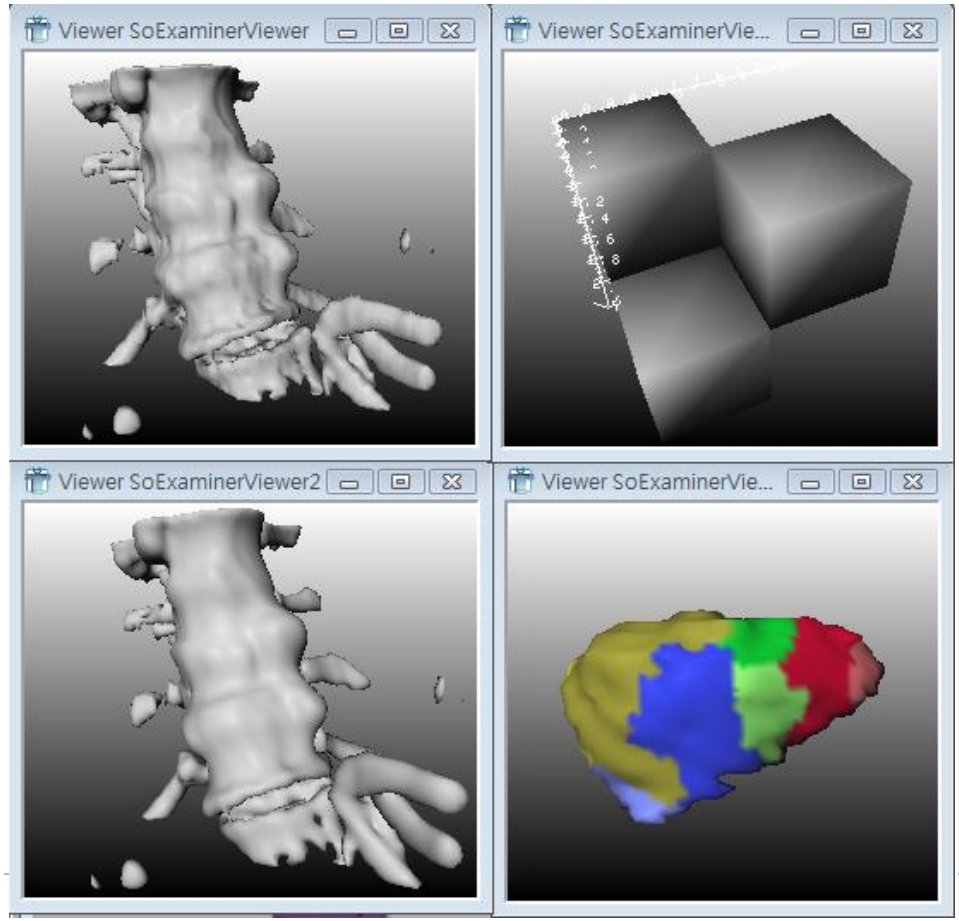


A liver surface colored by a LUT in bone context



Winged Edge Mesh IsoSurface

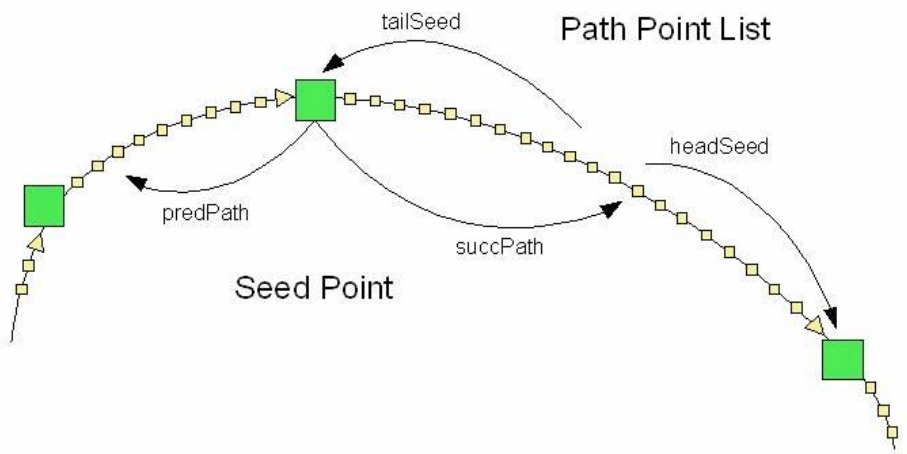
- Four subnetworks, each showing different features of the WEMIsoSurface



Contour Segmentation Objects (CSO)

- CSO library
  - provides data structures and modules for interactive or automatic generation of contours in voxel images
- Contours can be analyzed, maintained, grouped and converted back into a voxel image

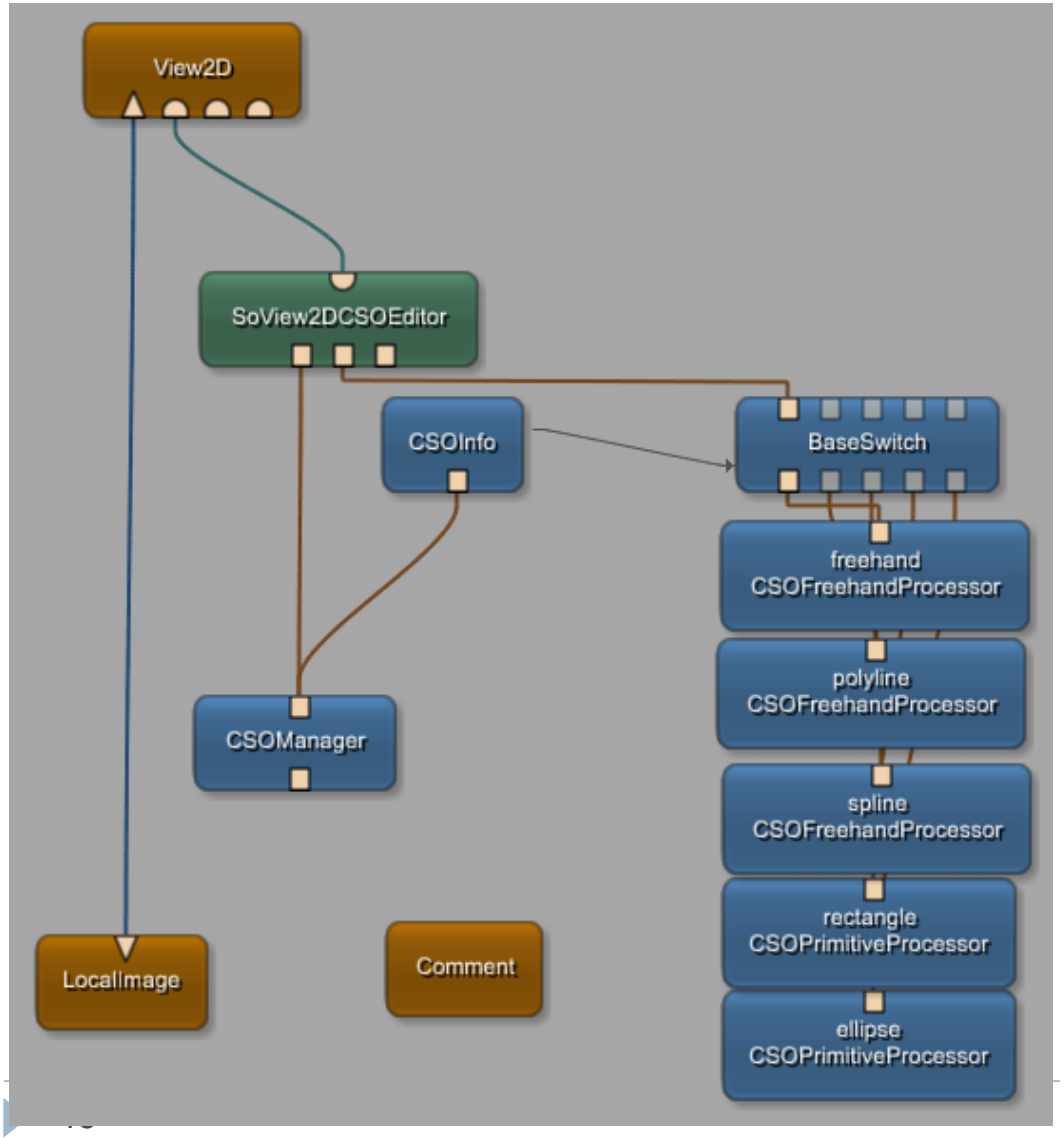
- CSO consists of a number of seed points and a number of path point lists

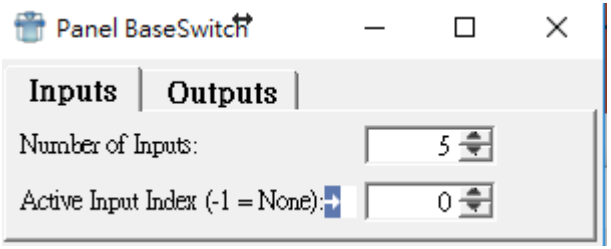


CSO Modules Overview

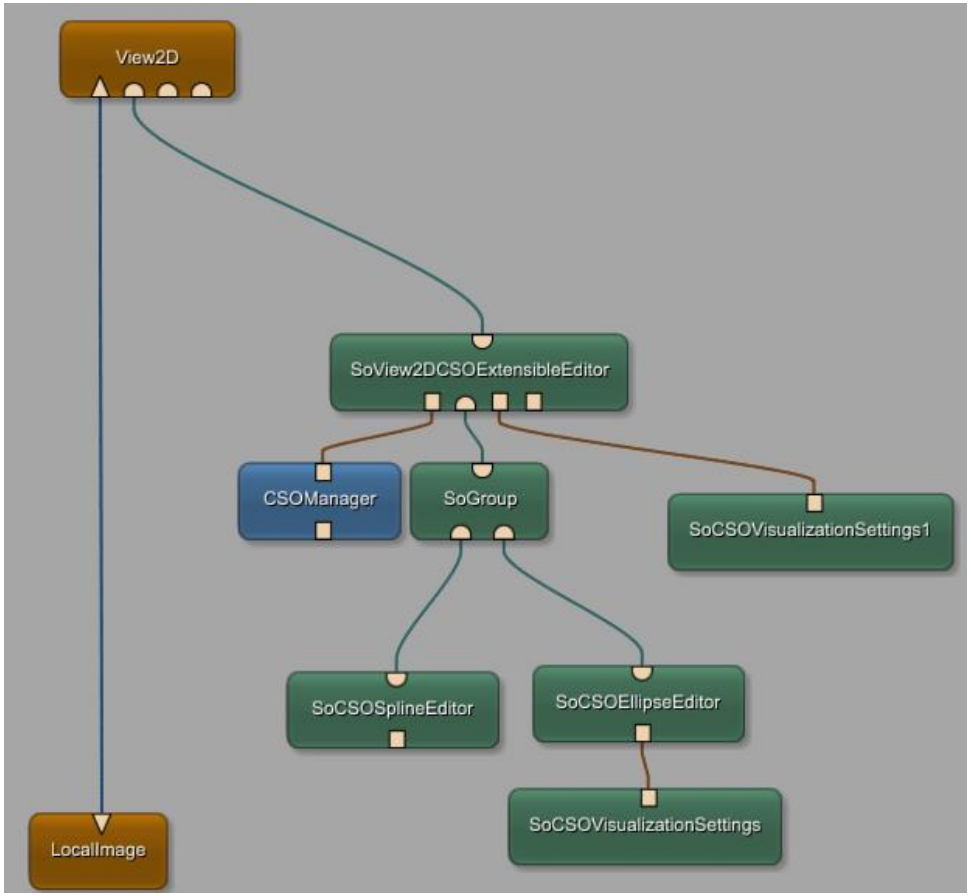
- Generation (without interaction):
  - CSOIsoGenerator
- Processing (with interaction):
  - CSOFreehandProcessor, CSOLiveWireProcessor, CSOIsoProcessor, CSOBulgeProcessor, …
- Rendering
  - SoView2DCSOEditor, SoCSO3DVis
- Misc
  - CSOConvertToImage, CSOConvertTo3DMask, CSOFilter, CSOManager, CSOLoad / CSOSave, …

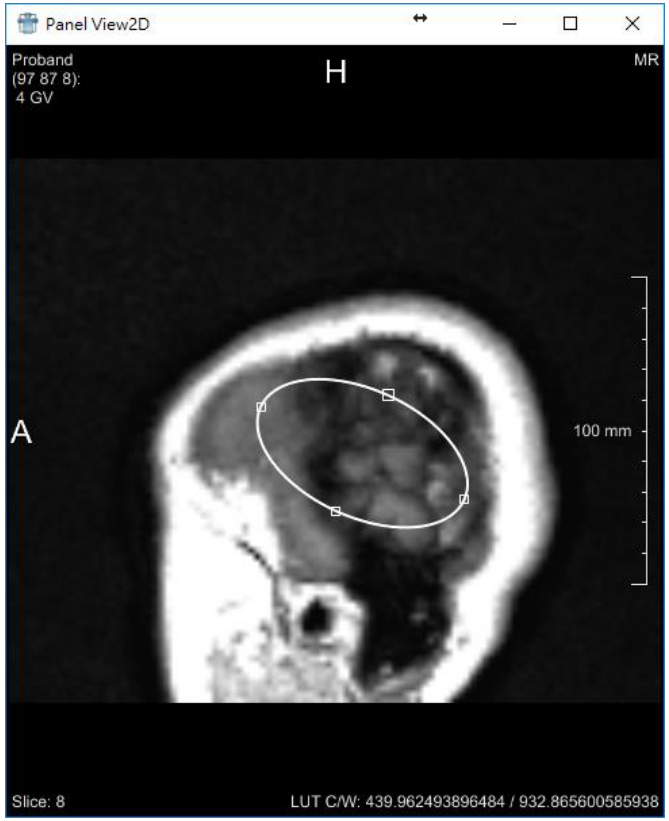
SoView2DCSOEditor Example Network



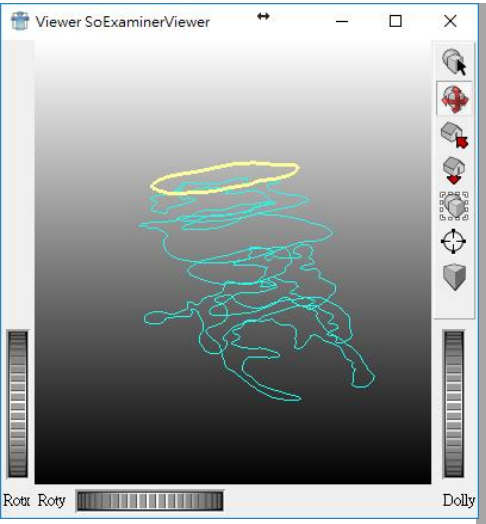


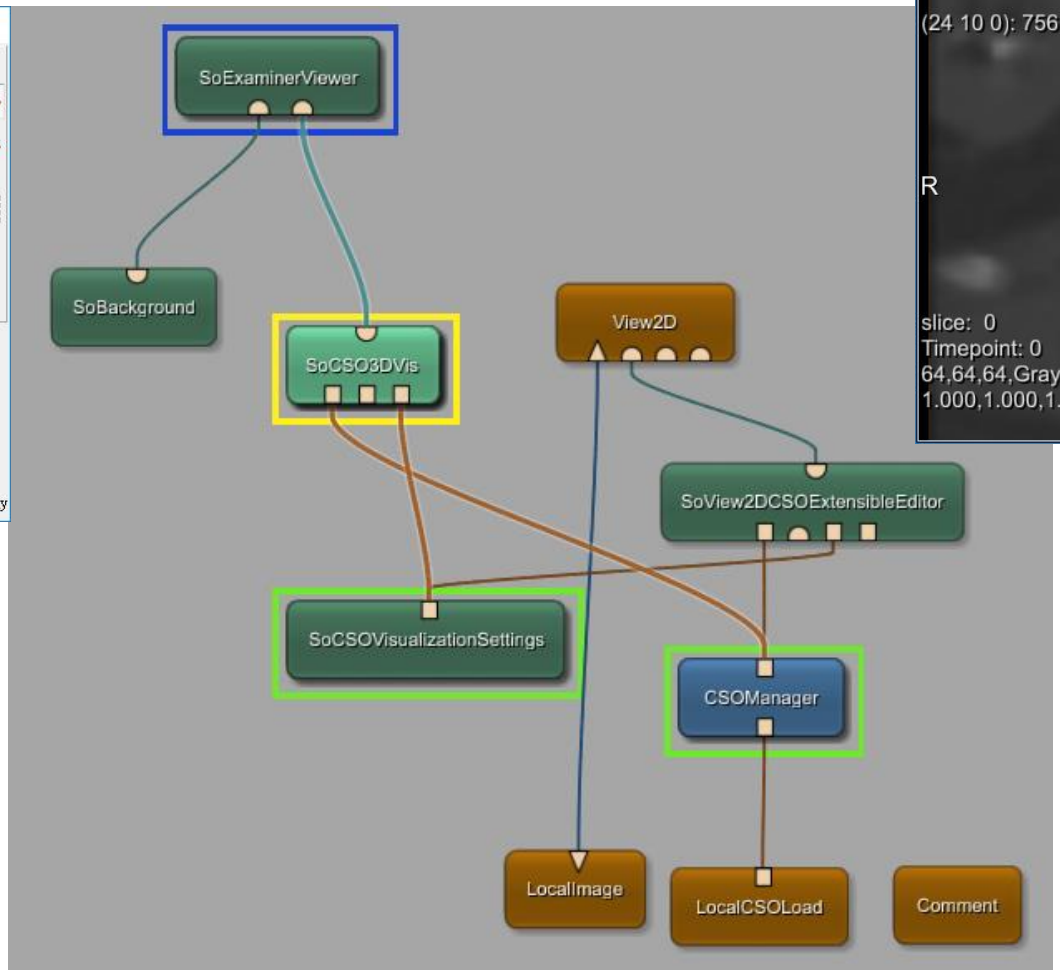
SoView2DCSOExtensibleEditor Example Network

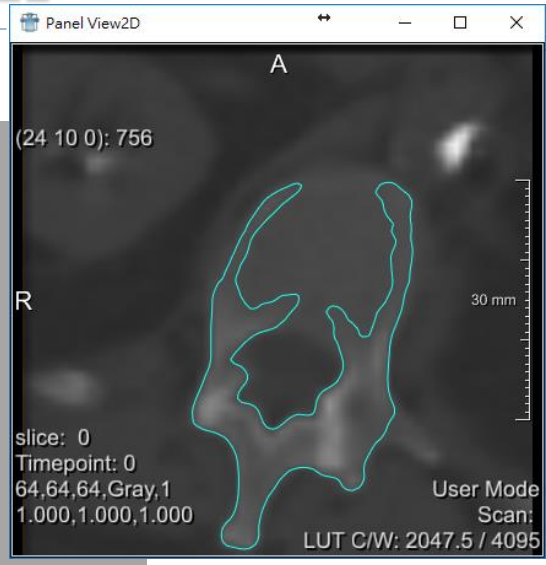




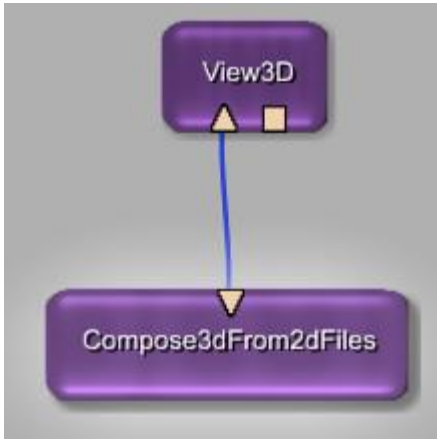
SoCSO3DVis Example Network

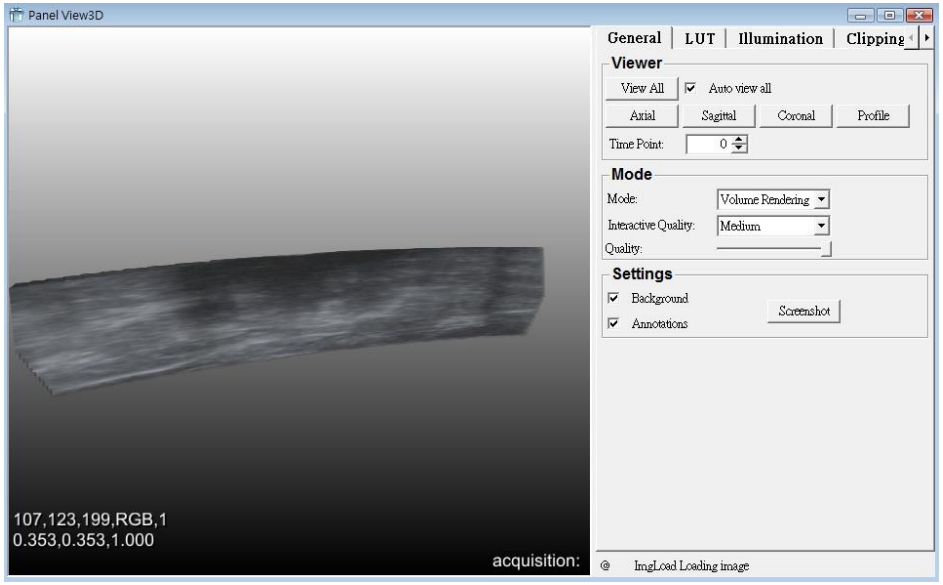


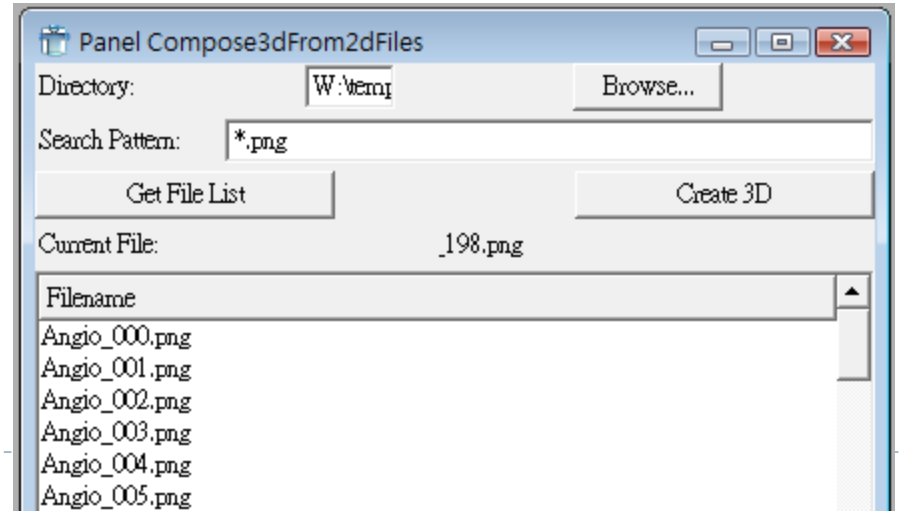


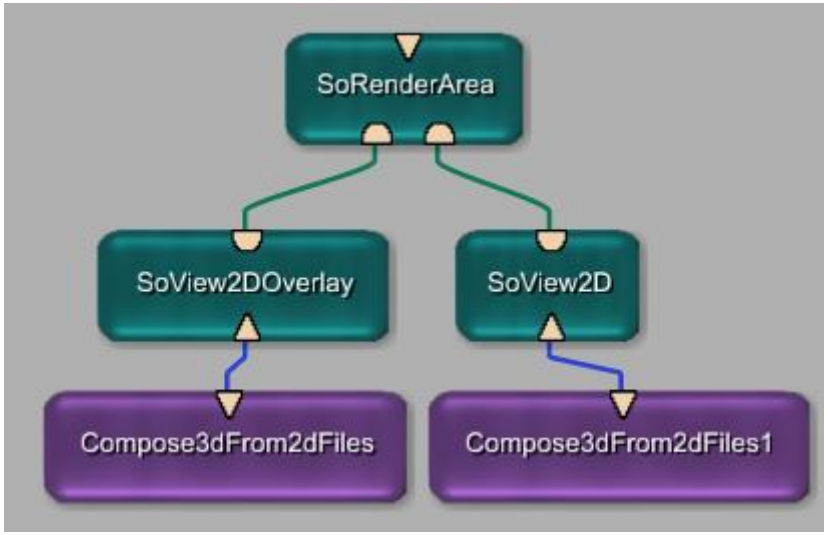


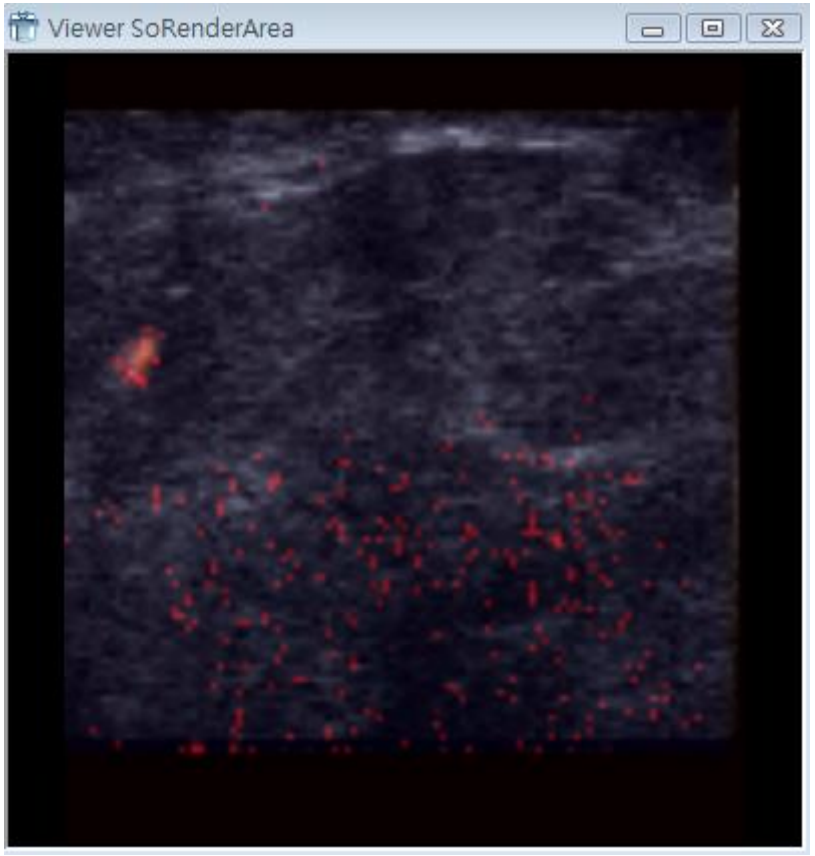
3D


View3D
Compose3dFrom2dFiles


Panel View3D
General
LUT
Illumination
Clipping
Viewer
View All
Auto view all
Axial
Sagittal
Coronal
Profile
Time Point:
0
Mode
Mode:
Volume Rendering
Interactive Quality:
Medium
Quality:
Settings
Background
Annotations
Screenshot
107,123,199,RGB,1
0.353,0.353,1.000
acquisition:
ImgLoad Loading image


Panel Compose3dFrom2dFiles
Directory:
W:\temp
Browse...
Search Pattern:
*.png
Get File List
Create 3D
Current File:
_198.png
Filename
Angio_000.png
Angio_001.png
Angio_002.png
Angio_003.png
Angio_004.png
Angio_005.png


SoRenderArea
SoView2DOverlay
SoView2D
Compose3dFromFiles
Compose3dFromFiles1


Viewer SoRenderArea

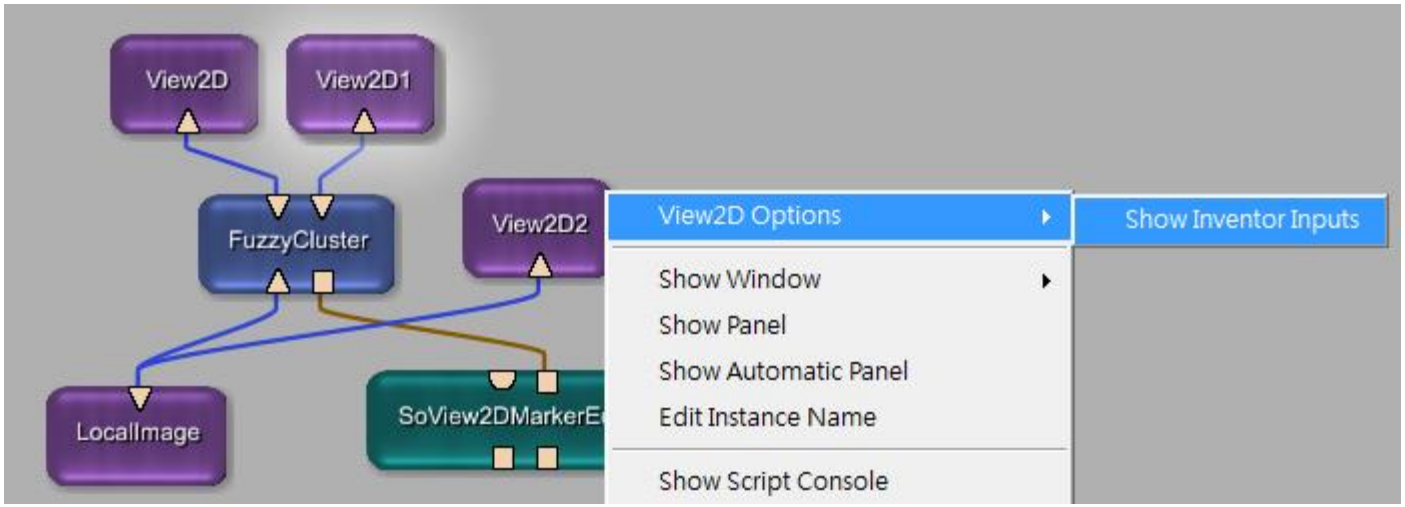
DICOM Support

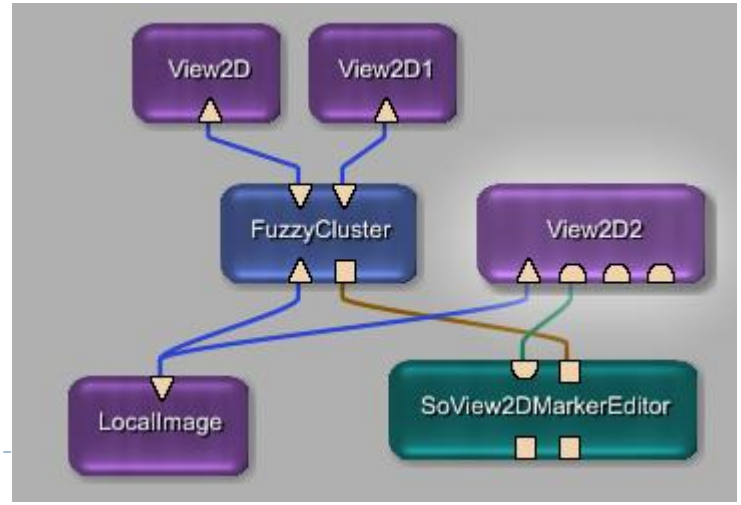
- Import of 2D/3D/4D DICOM datasets
- MeVisLab DICOM Service runs as Windows Service or UNIX Daemon and receives data from PACS even when user is logged out
- Export of DICOM slices to disk
- DICOM-Store allows to send data to PACS

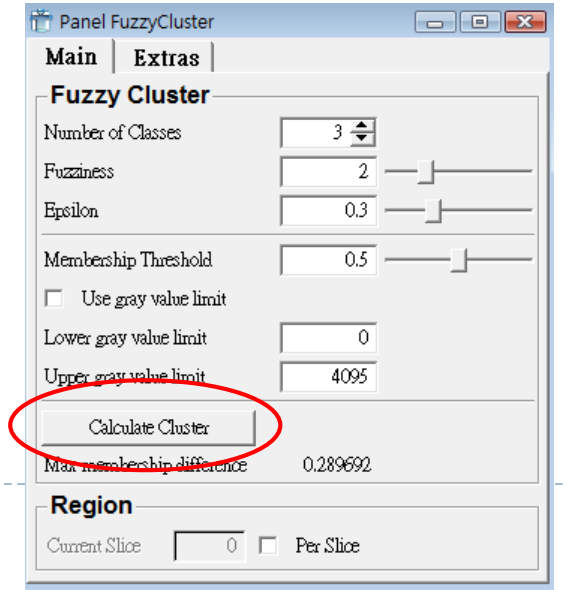
Fuzzy

- FuzzyCluster
  - an implementation of the fuzzy c-means algorithm that classifies an image into different clusters depending on the gray values
- FuzzyConnectDistance
  - a segmentation algorithm based on Fuzzy Connectedness extended by the possibility to use a property based on the distance of image elements to the center of the object to be segmented while calculating membership values
- FuzzyObjectLabeling

FCM


View2D
View2D1
FuzzyCluster
View2D2
LocalImage
SoView2DMarkerE
View2D Options
Show Inventor Inputs
Show Window
Show Panel
Show Automatic Panel
Edit Instance Name
Show Script Console


View2D
View2D1
FuzzyCluster
View2D2
LocalImage
SoView2DMarkerEditor


Panel FuzzyCluster
Main
Extras
Fuzzy Cluster
Number of Classes
3
Fuzziness
2
Epsilon
0.3
Membership Threshold
0.5
Use gray value limit
Lower gray value limit
0
Upper gray value limit
4095
Calculate Cluster
0.289692
Region
Current Slice
0
Per Slice


Panel View2D2
Panel View2D1
Panel View2D
SANKT-JUERGEN-STR., BREMEN
29990101 F MAGNETOM VISION
id0
MR
(255 77 0):
(248 65 0):
(223 10 0):
(12 23 0):
(134 20 0):
(8 63 0):
(255 201 0):
(252 208 0):
(200 204 0):
0 GV
250 GV
A
Slice: 0
Timepoint: 0
256,256,3,Grey,1
1.000,1.000,1.000
MR Mode
ILAB4
Scan: 20000311
LUT C/W: 188.681 / 250
LUT C/W: 1 / 4

ITK Wrapper

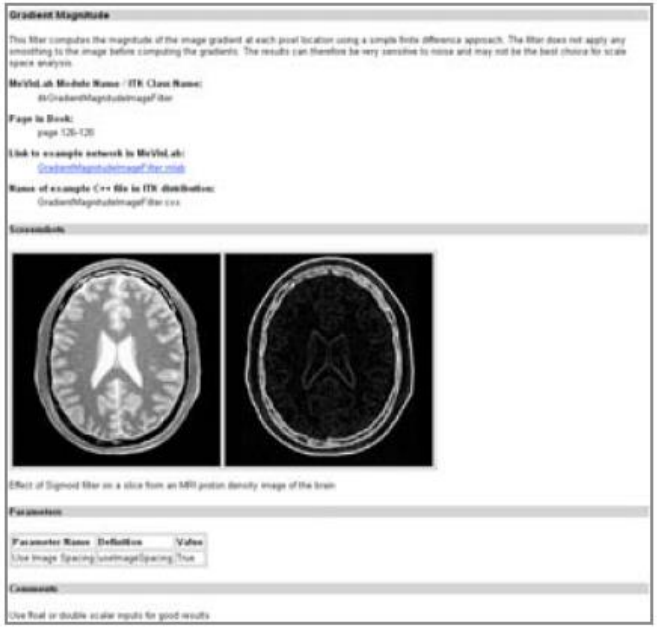
- ITK – Insight Toolkit (www.itk.org)
  - Open Source Library for Medical Image Processing and Registration
- about 200 Modules for Standard Image Processing such as
  - Image Arithmetics
  - Kernel-based and Diffusion Filtering
  - Levelset and Segmentation Filtering
  - Warping, Resampling Filters
- about 90 Modules Registration-Related Algorithms
  - Interpolators
  - Metrics
  - Optimizers
  - Transformations

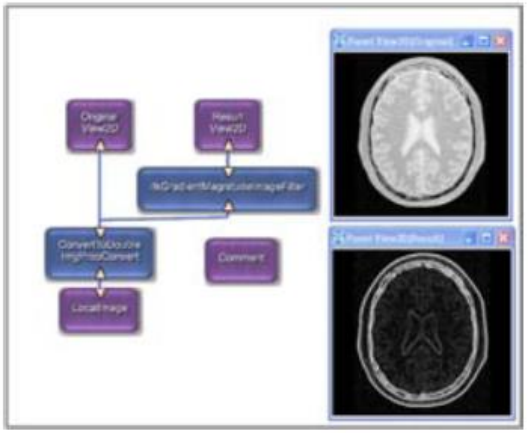
ITK Book Examples

ITK Book Example → Corresponding Website (screenshots generated with MeVisLab) → MeVisLab Network

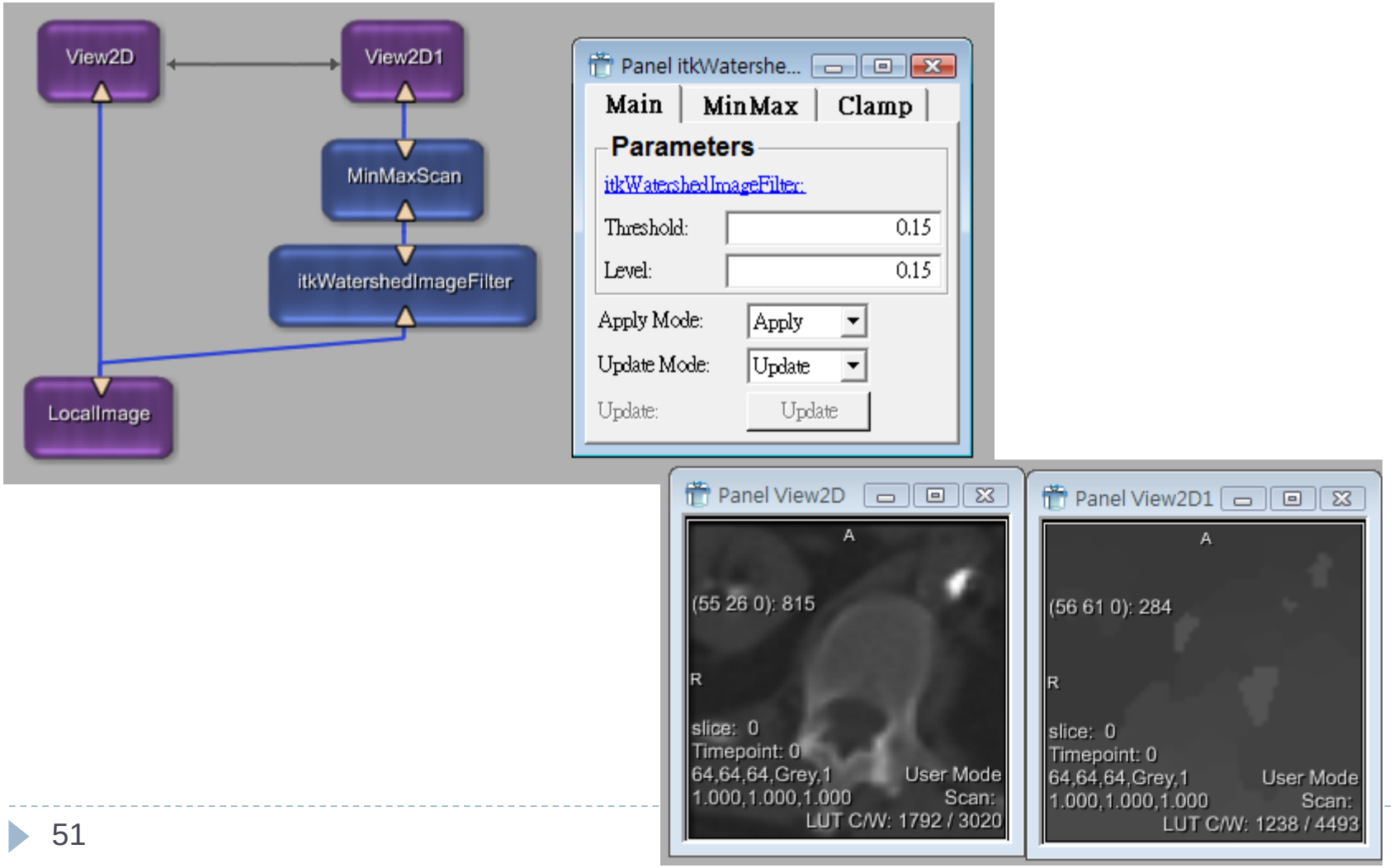
www.itk.org/ItkSoftwareGuide.pdf

www.mevislab.de/index.php?id=35

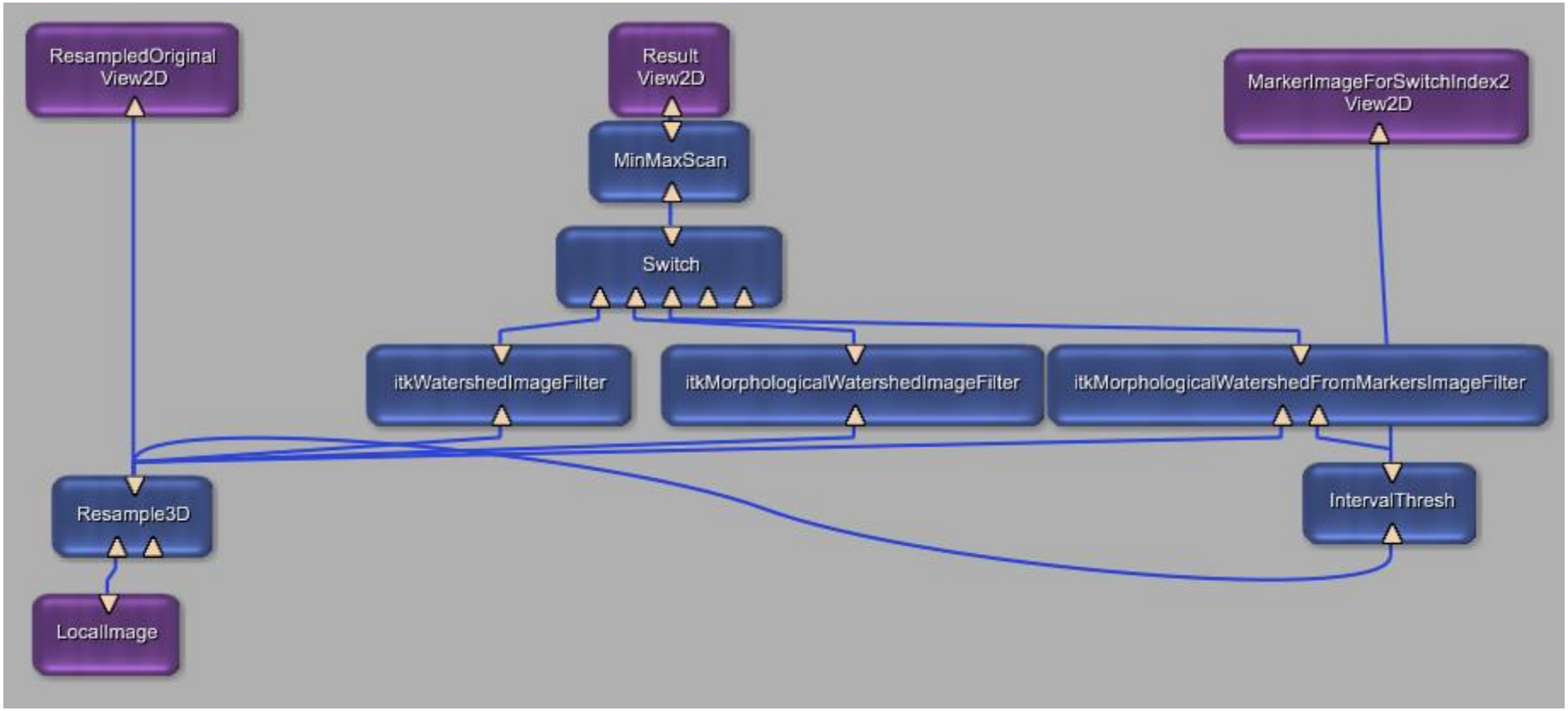


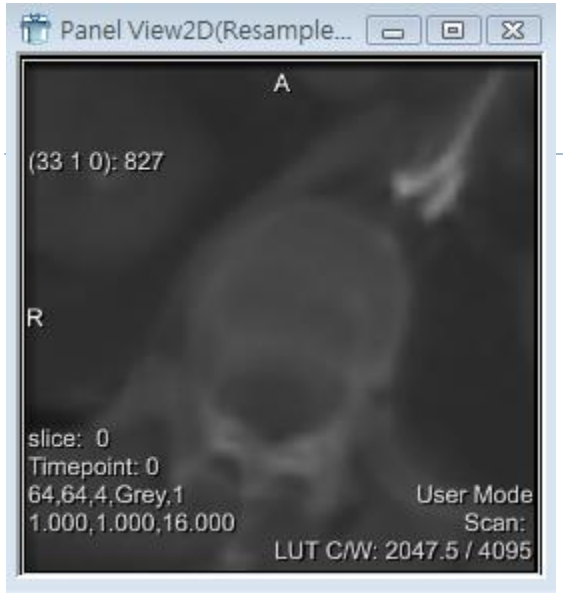


ITK Watershed

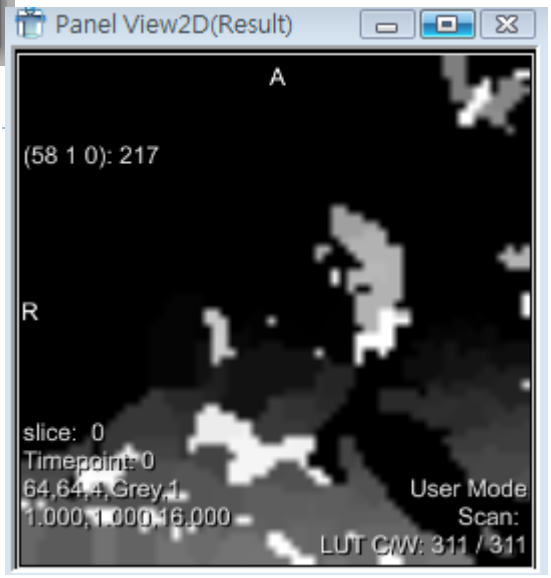


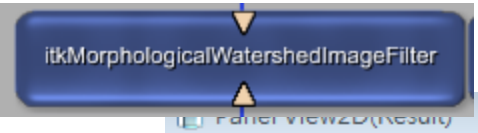
Example Network of itkWatershedImageFilter

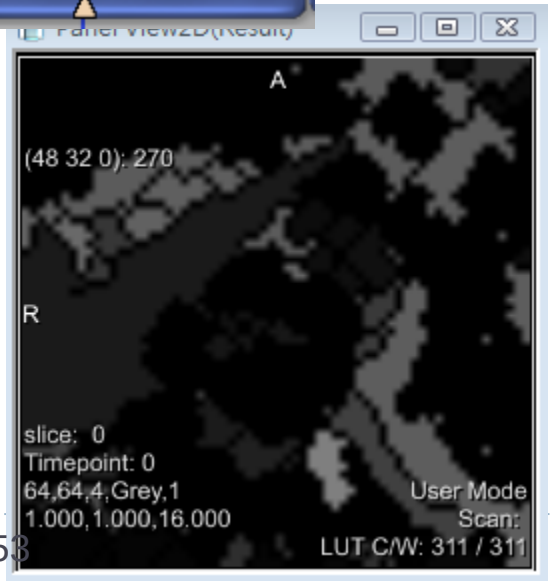



Panel View2D(Resample...
A
(33 1 0): 827
R
slice: 0
Timepoint: 0
64,64,4,Grey,1
1.000,1.000,16.000
User Mode
Scan:
LUT C/W: 2047.5 / 4095

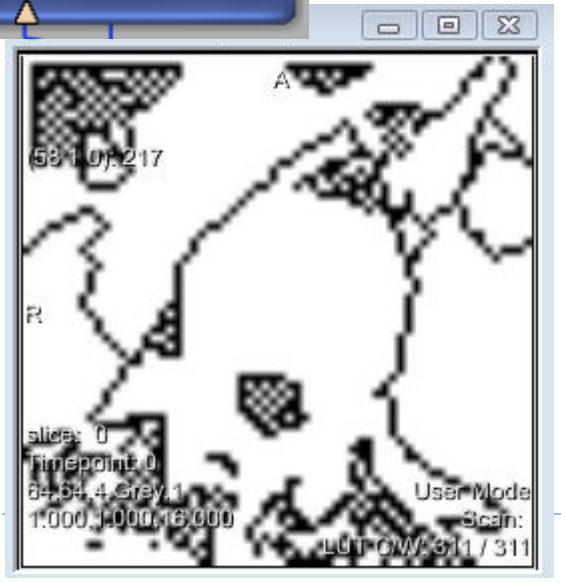

itkWatershedImageFilter


Panel View2D(Result)
A
(58 1 0): 217
R
slice: 0
Timepoint: 0
64,64,4,Grey,1
1.000,1.000,16.000
User Mode
Scan:
LUT C/W: 311 / 311


itkMorphologicalWatershedImageFilter


A
(48 32 0): 270
R
slice: 0
Timepoint: 0
64,64,4,Grey,1
1.000,1.000,16.000
User Mode
Scan:
LUT C/W: 311 / 311

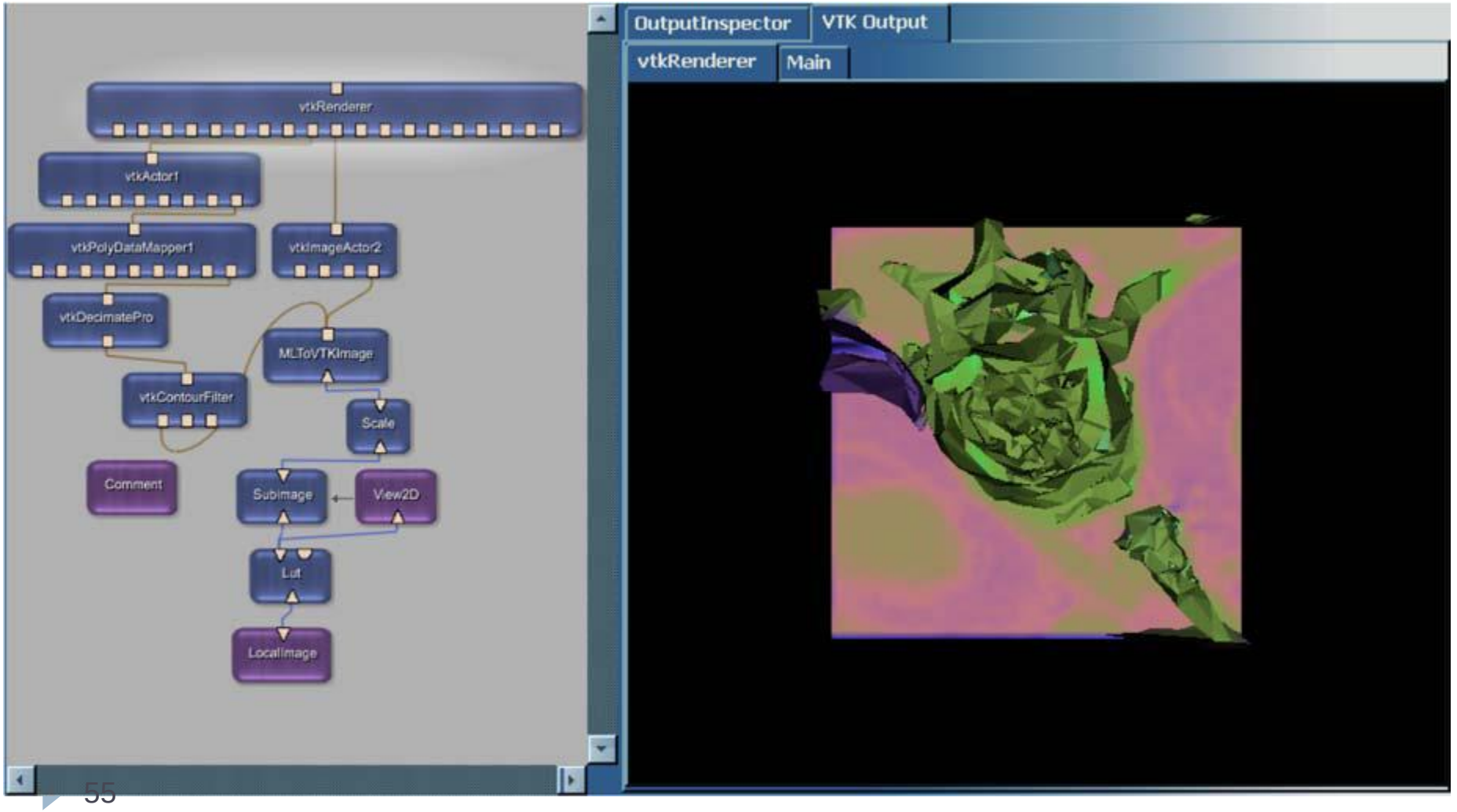

itkMorphologicalWatershedFromMarkersImageFilter


A
R
User Mode
Scan:
LUT C/W: 311 / 311

VTK Wrapper

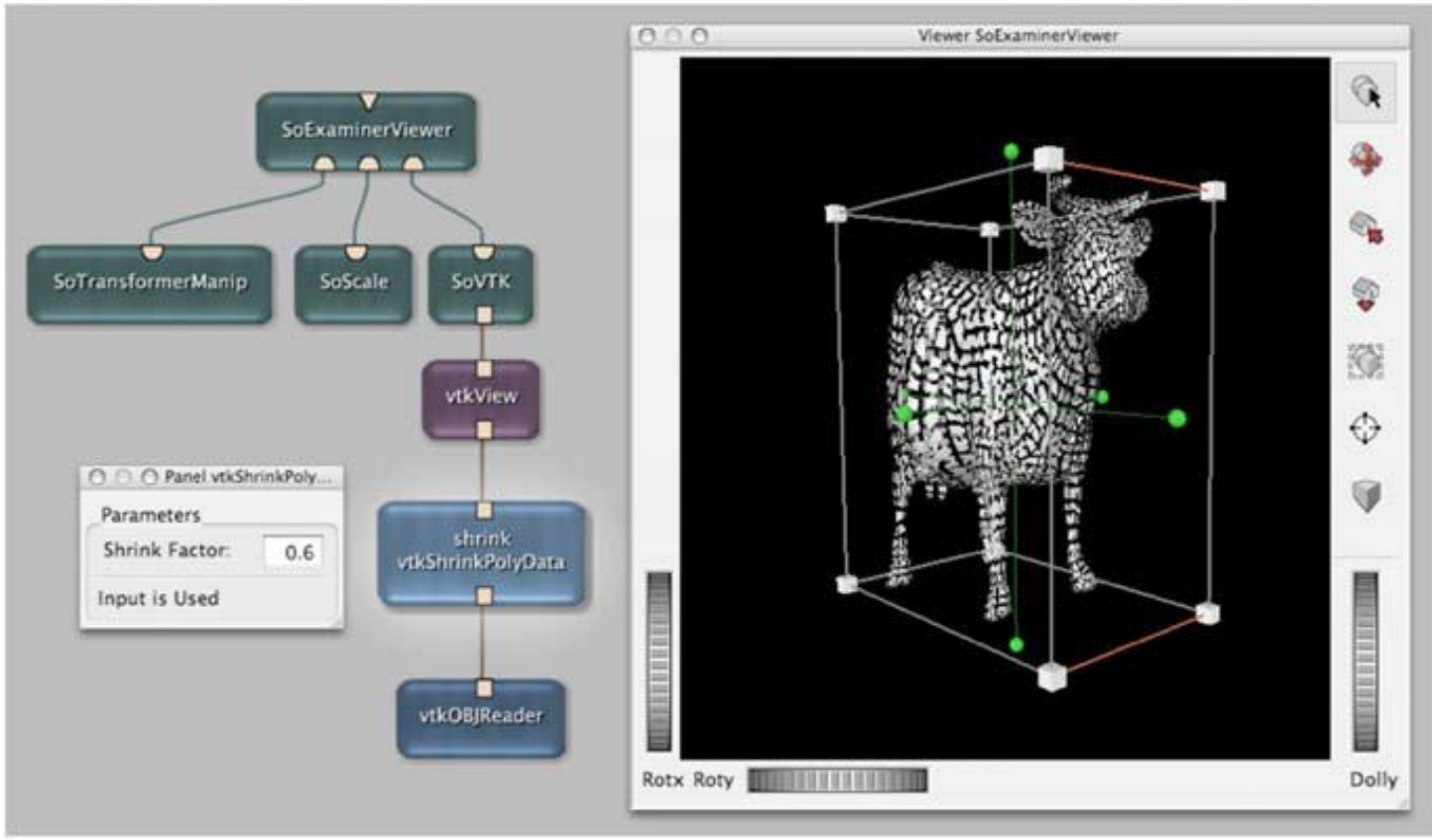
- VTK – Visualization Toolkit (www.vtk.org)
  - Visualization, Image Processing and Filtering Library for images, meshes, grids, data sets etc.
- about 1000 Modules for
  - 2D/3D Image Processing
  - Grid, Mesh, Surface, and Data Filtering
  - Pickers
  - Properties and Actors
  - Mappers
  - Renderers, Widgets, Viewers
  - Sources, Readers and Writers
  - Transformations

VTK Example 1: Contour Filter

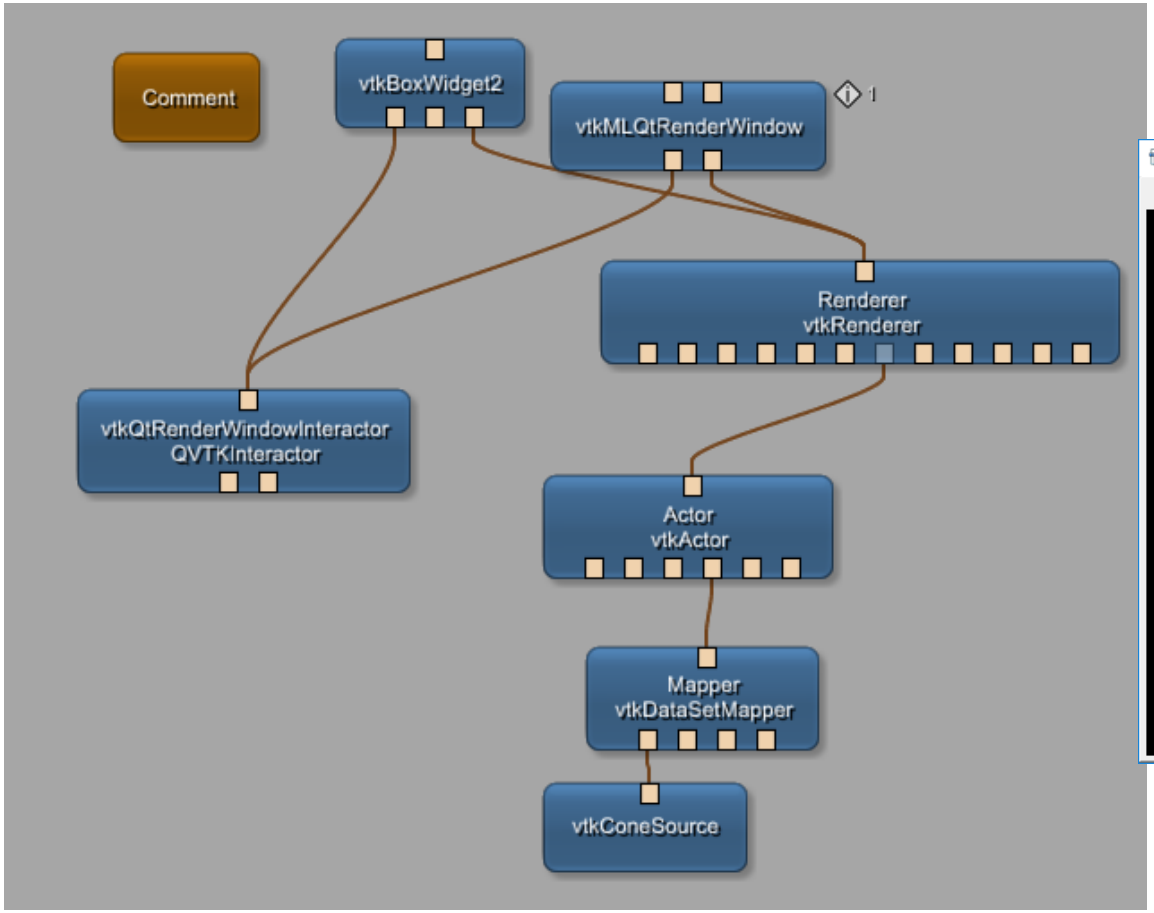


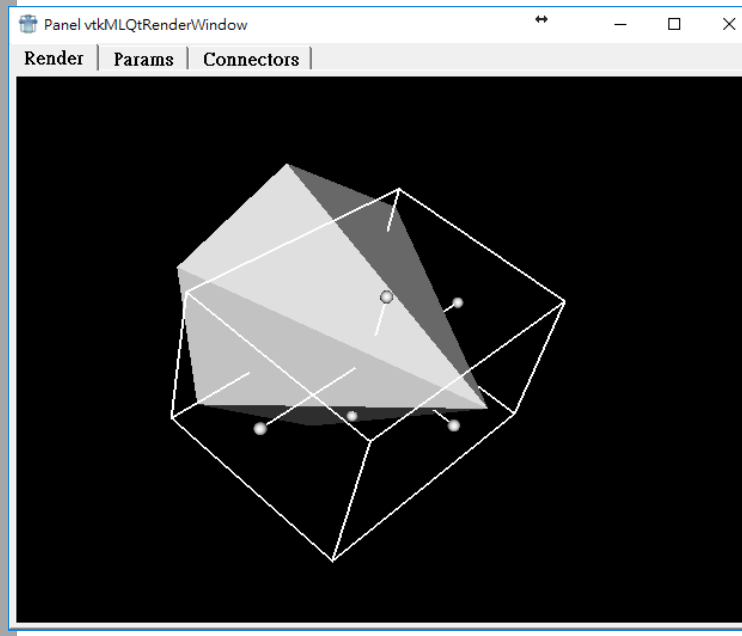
VTK Example 2: VTK/OIV mix

- SoVTK module allows VTK rendering as part of an Open Inventor scene graph

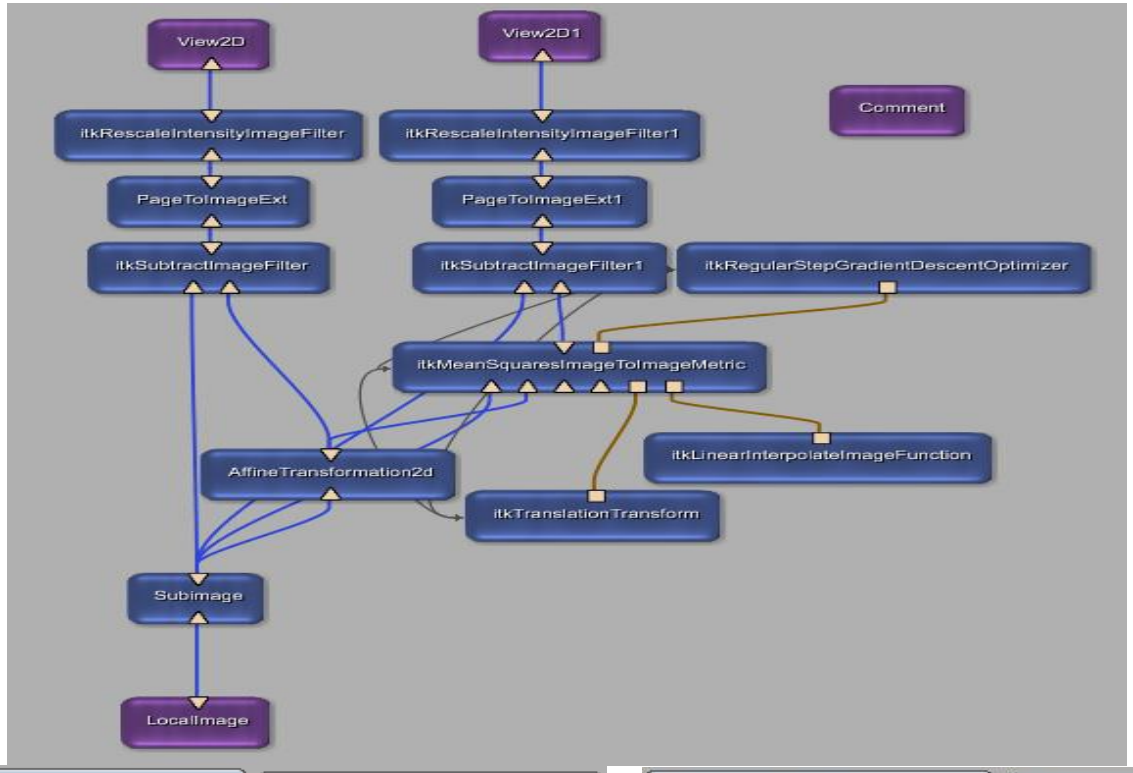


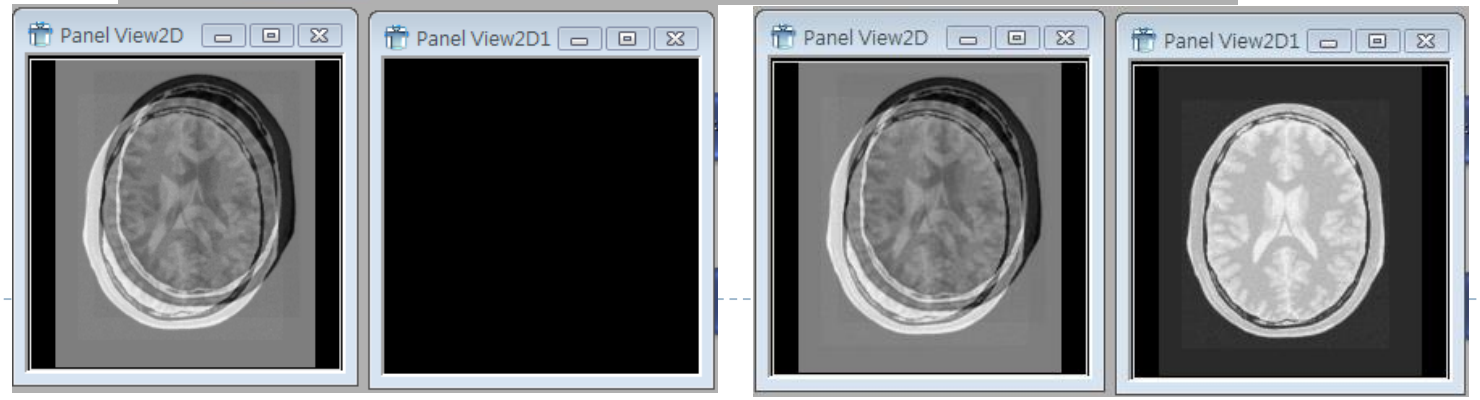
vtkBoxWidget2 Example Network





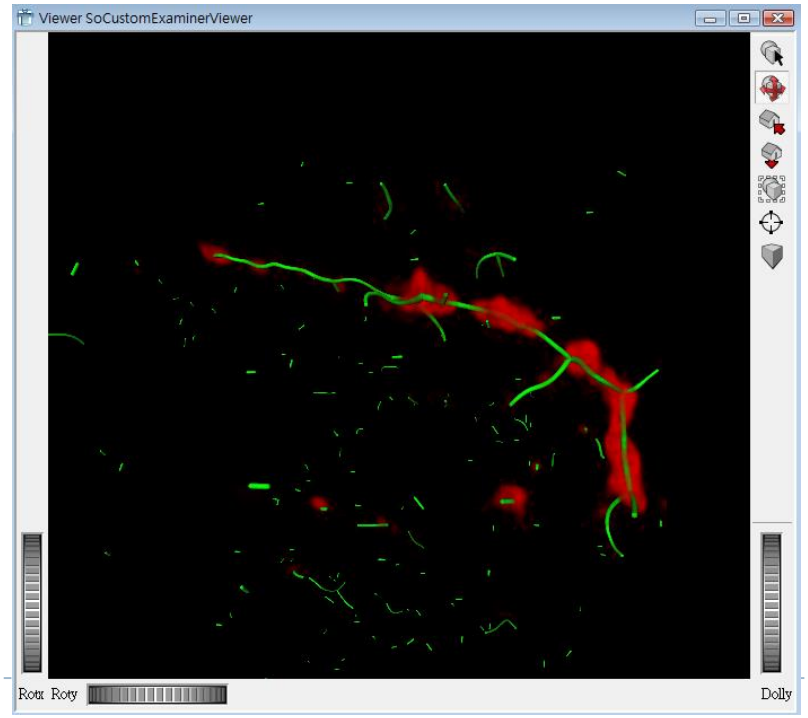
ITK Image Registration

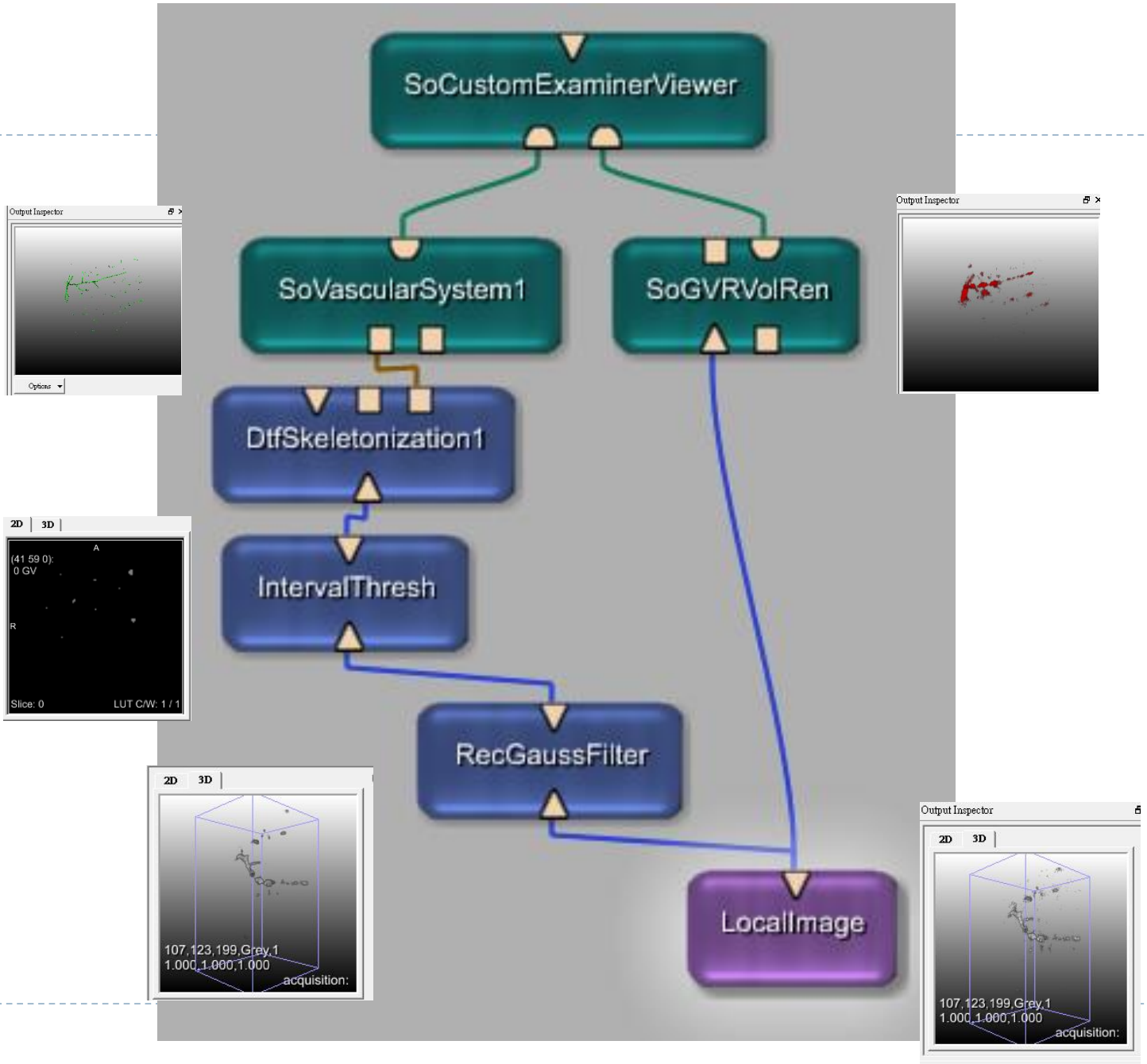




Skeletonization

- Skeletonization of a binary image by successive erosion of border voxels
  - vessel centerline extraction
  - in 2d and 3d




SoCustomExaminerViewer
SoVascularSystem1
SoGVRVolRen
DtfSkeletonization1
IntervalThresh
RecGaussFilter
LocalImage
Output Inspector
Options
Output Inspector
2D
3D
(41 59 0):
0 GV
Slice: 0
LUT C/W: 1 / 1
2D
3D
107,123,199,Grey,1
1.000,1.000,1.000
acquisition:
Output Inspector
2D
3D
107,123,199,Grey,1
1.000,1.000,1.000
acquisition: