Visual Medicine: Part one - Foundations of Medical Imaging

Visual Programming for Prototyping of Medical Imaging Applications

Florian Link

MeVis Research
Bremen, Germany

link@mevis.de

Overview

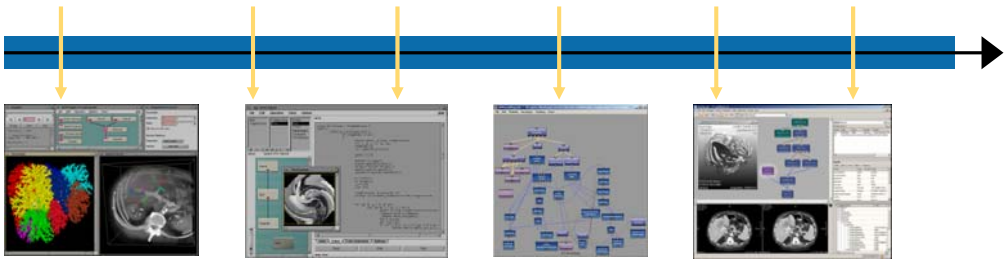
- Introduction to MeVisLab
- Visual Programming
- Examples
- VTK / ITK Integration
- MeVisLab SDK Features

MeVisLab

MeVisLab is:

- MeVis Research and Development Platform
- Medical Image Processing and Visualization
- Rapid Application Prototyping

History



1993: ILAB1

– 1997: ILAB2 & 3

2000: ILAB4

2002: Me*VisLab*

2004: www.mevislab.de

Licensing

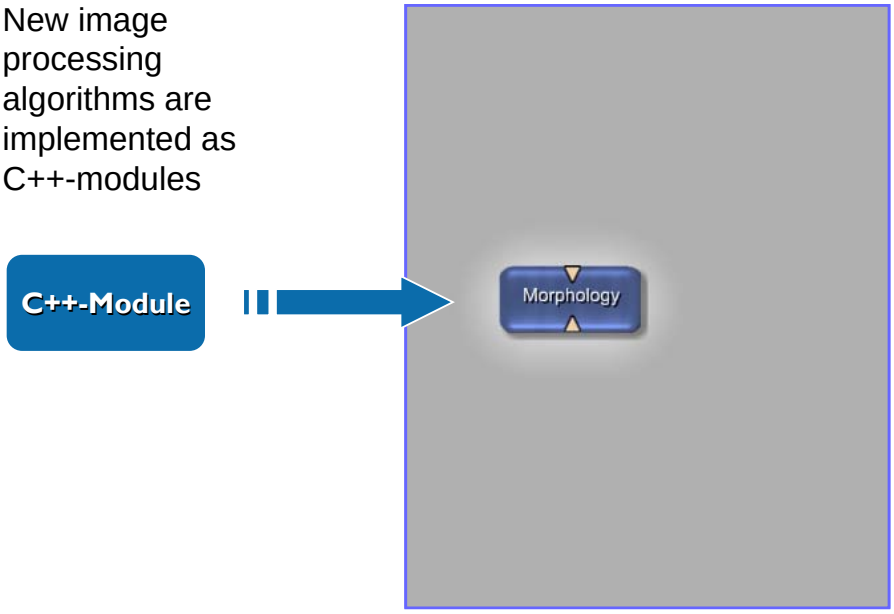
- **MeVisLab Basic** is free for non-commercial usage
- Many algorithms presented in this tutorial can be explored with **MeVisLab Basic**
- **Full MeVisLab SDK** is available at academic and commercial rates
  - 3 month evaluation version available

Other Visualization Platforms

- Amira
- Analyze
- AVS Express
- IBM Data Explorer/OpenDX
- Khoros/VisQuest
- LONI
- SCIRun
- …

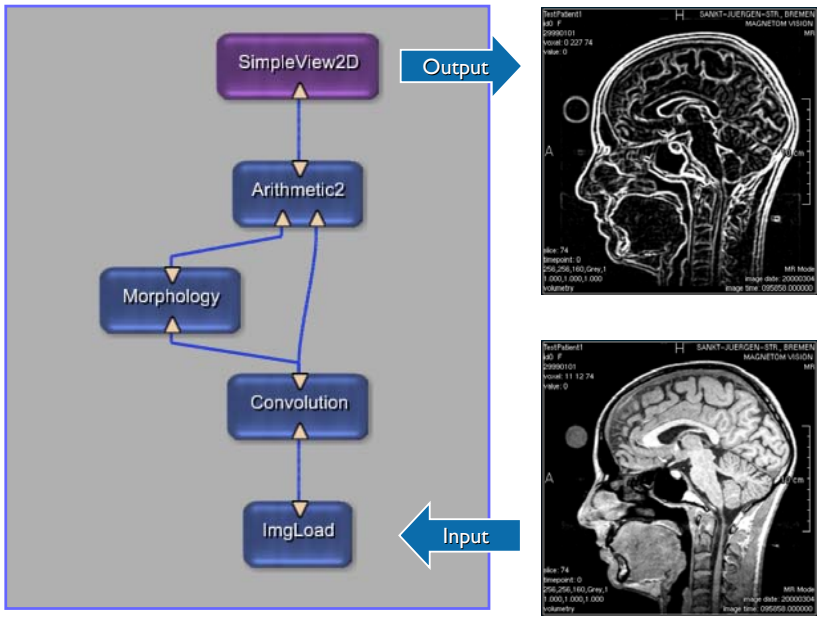
Visual Programming

New image processing algorithms are implemented as C++-modules



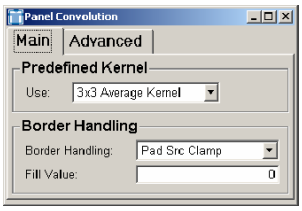
Visual Programming

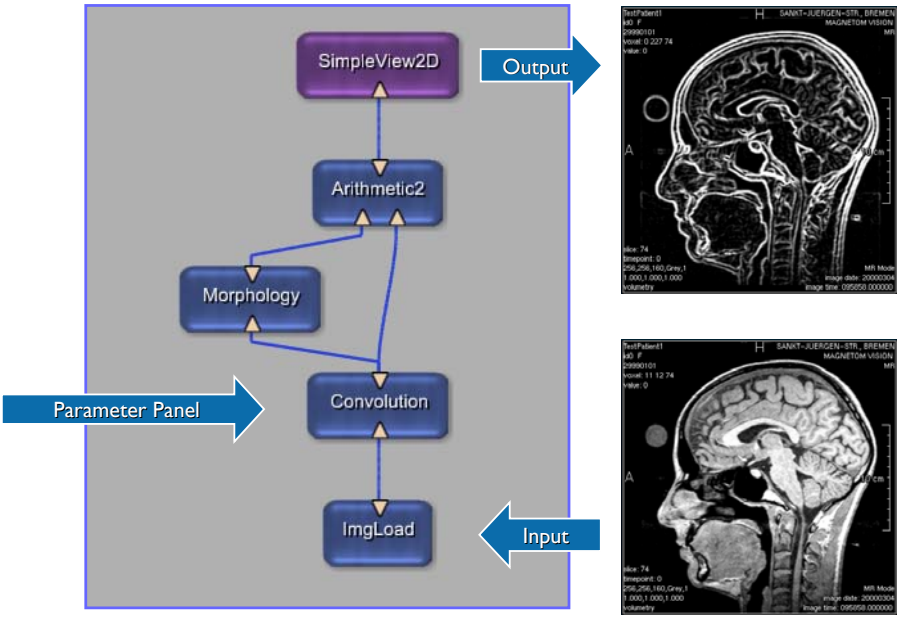
Individual image processing modules are combined to networks using a graphical user interface



Visual Programming

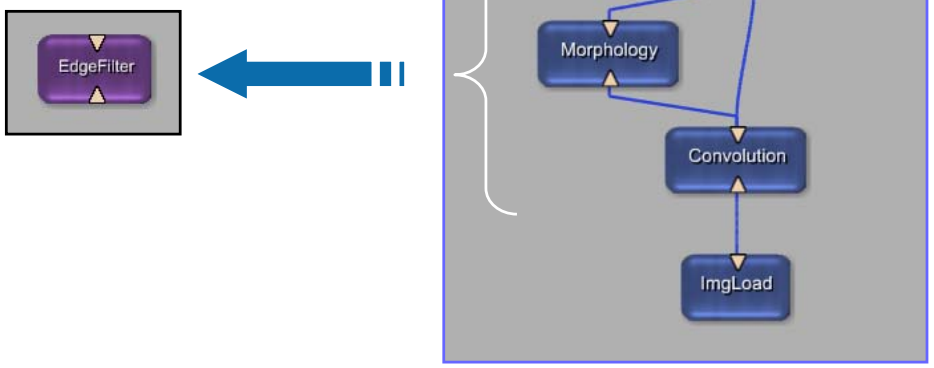
Each image processing module can be controlled using its own parameter panel



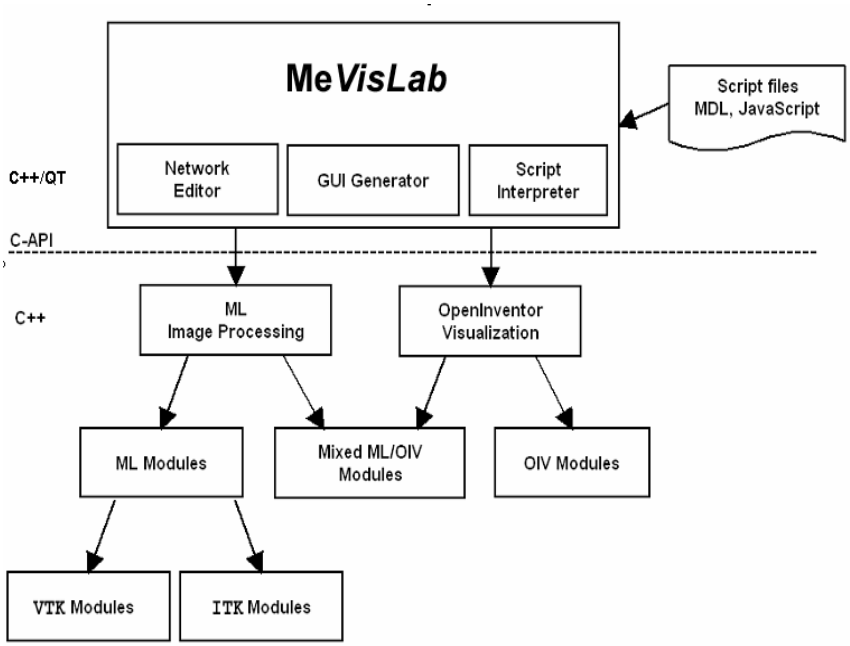


Macro Modules

- Encapsulate subnetworks
- Define interfaces
- Scripting



MeVisLab Architecture



Available Modules

- 450 Image Processing Modules
- 300 Open Inventor Modules
- 400 Macro Modules
- 280 ITK Modules
- 740 VTK Modules

Over 40 module developers (including students)

Image Processing

- MeVis Image Processing Library (ML)
- Page oriented and request driven
- Priority controlled caching
- General image concept:
  - Various data types (int, float, complex, tensors)
  - x/y/z/c/t/u dimensions
- Medical image properties:
  - DICOM coordinate system and tags
- C++ Interface and Wizard available for integration of new algorithms
- Detailed user guide at www.mevislab.de

Image Processing

Filters
- Diffusion filters
- Morphology filters
- Kernel filters

Segmentation
- Region growing
- Live wire
- Fuzzy connectedness
- Threshold
- Manual contours

Transformations
- Affine transformations
- Distance transformations
- Radon transform
- Manual registration

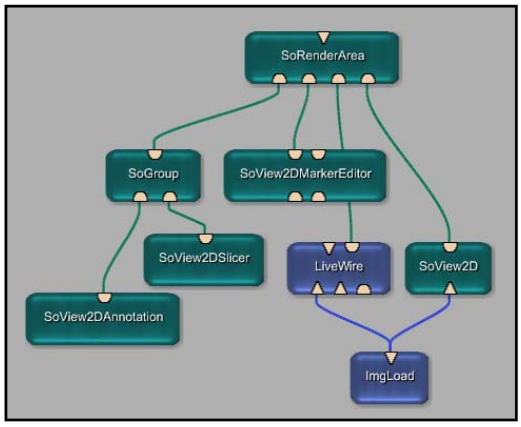
Statistics
- Histograms
- Global image statistics
- Box counting dimension

Other
- Unary/binary arithmetic
- Resampling/reformatting
- Oblique and curved MPR
- Dynamic data analysis
- Noise/test pattern generators

Open Inventor

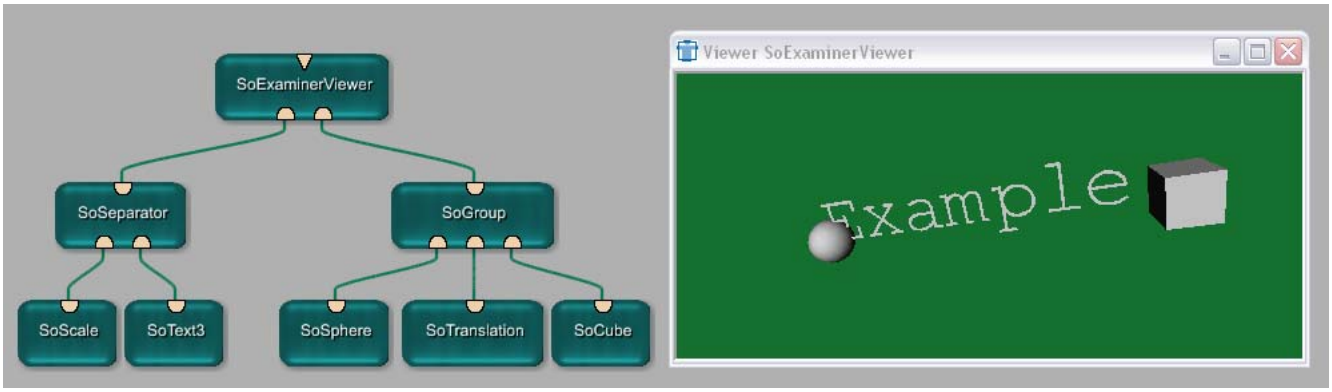
- Direct OpenInventor node support
- OpenInventor:
  - Scene graph paradigm
  - Object, rendering, transformation, property, … nodes
  - Based on OpenGL
  - Well documented
- Specialized interface for 2D viewer extensions
- Mixed ML/OpenInventor-modules

www.mevislab.de/inventor



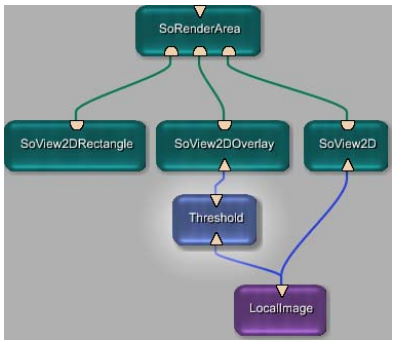
Open Inventor Scene Graph

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



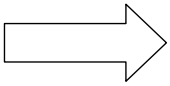
2D Viewers

- 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



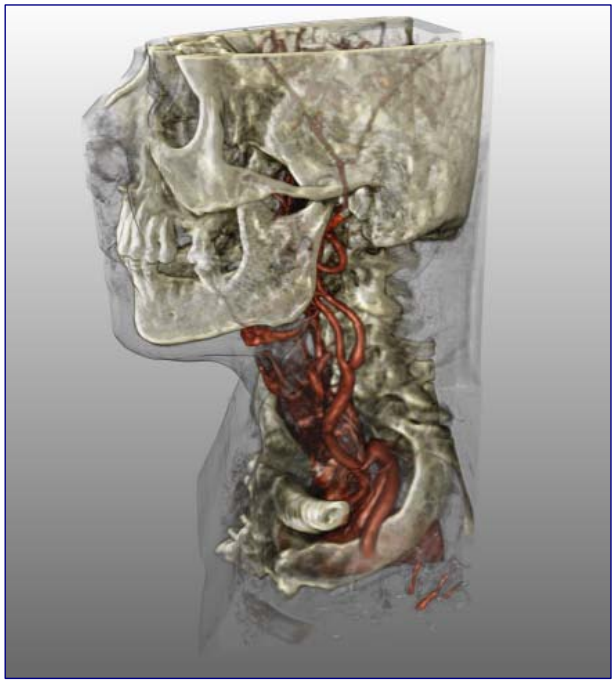
Volume Rendering

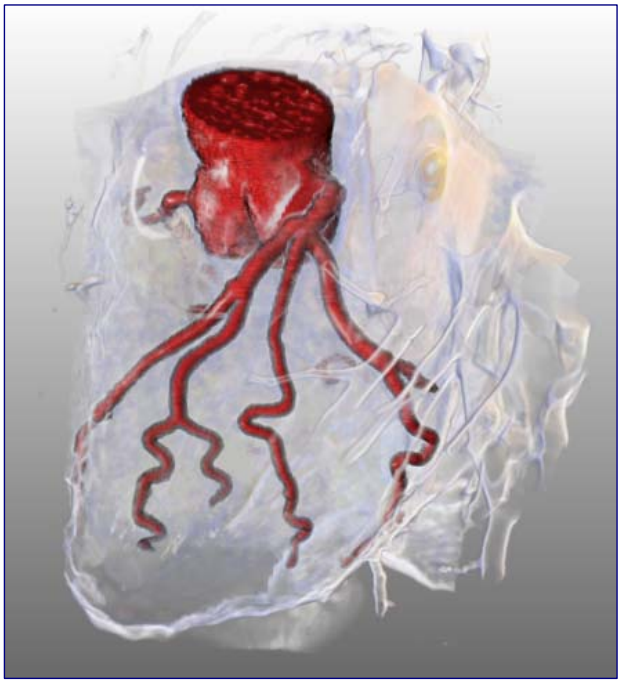
- Advanced Volume Rendering modules
  - MIP, DVR, Shaded DVR
  - Tone Shading, Silhouette and Boundary Enhancement
  - Tagged/Labeled Objects
  - Per Object Shading
  - Large data visualization via multi-resolution data octree



See later talk for details!

Volume Rendering Examples



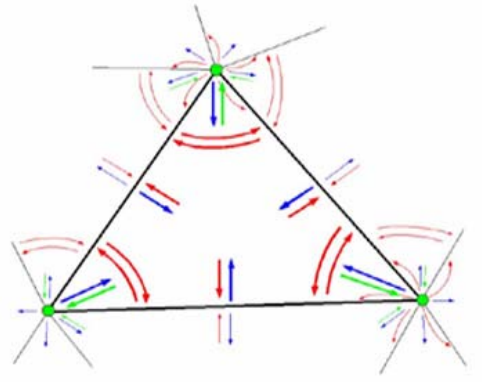


Winged Edge Mesh Library (WEM)

WEM: Winged-Edge Mesh

- 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

Rendering:

- SoWEMRenderer
  - Different Render Modes
  - Optional Coloring by LUT Values

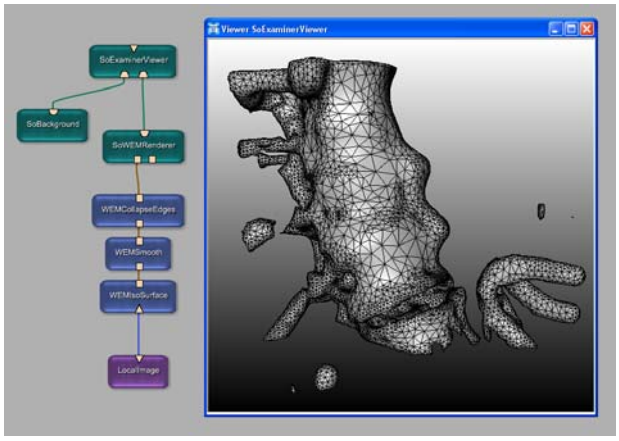
Processing:

- WEMCollapseEdges
- WEMSmooth
- WEMPurge
- WEMClip

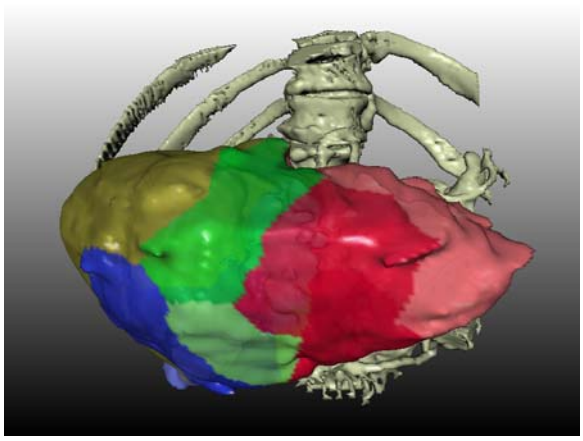
… and many more, type in ‘WEM’ in the search field.

WEM Screenshots

Network with iso surface generation and polygon reduction.



A liver surface colored by a LUT in bone context.



DICOM Support

- Import of 2D/3D/4D DICOM datasets
- MeVisLab DICOM Service runs as NT Service 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

ITK AddOn

ITK – Insight Toolkit (www.itk.org)

Open Source Library for Medical Image Processing and Registration

> 185 Modules for Standard Image Processing such as

- Image Arithmetics
- Kernel-based and Diffusion Filtering
- Levelset and Segmentation Filtering
- Warping, Resampling Filters

> 65 Modules Registration-Related Algorithms

- Interpolators
- Metrics
- Optimizers
- Transformations

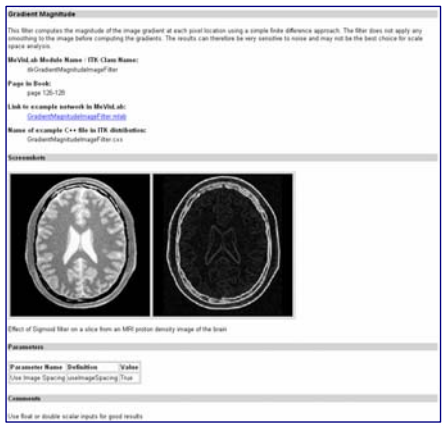
A few hundred other classes such as functions etc.

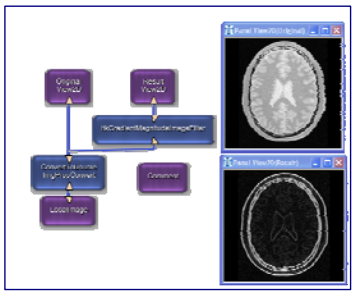
ITK Book Examples

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

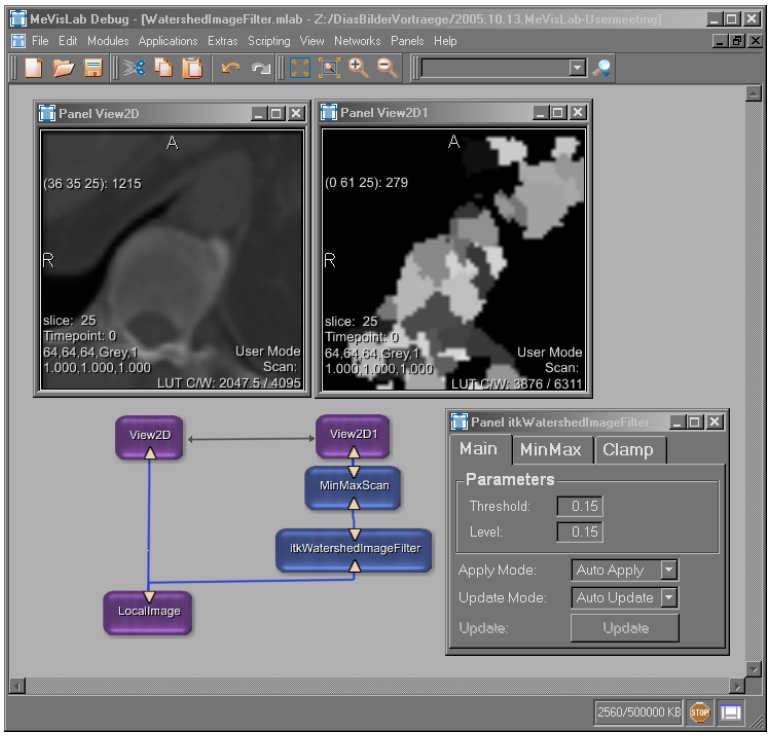
www.itk.org/ItkSoftwareGuide.pdf

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





ITK Example



Smooth integration with ML image processing

⇒ ITK modules behave like normal ML modules

Each filter has additional controls for

- Clamping of image values
- Min / Max setting
- Update / Apply handling

VTK AddOn

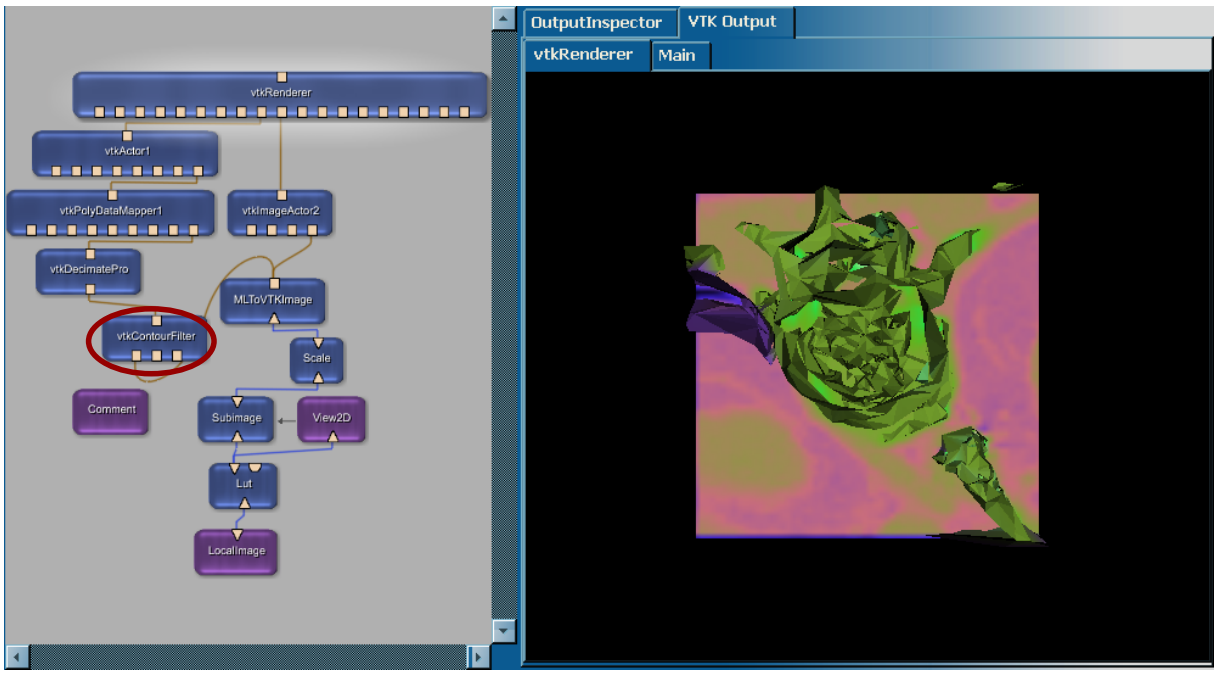
VTK – Visualization Toolkit (www.vtk.org)

Visualization, Image Processing and Filtering Library for images, meshes, grids, data sets etc.

> 700 Modules for

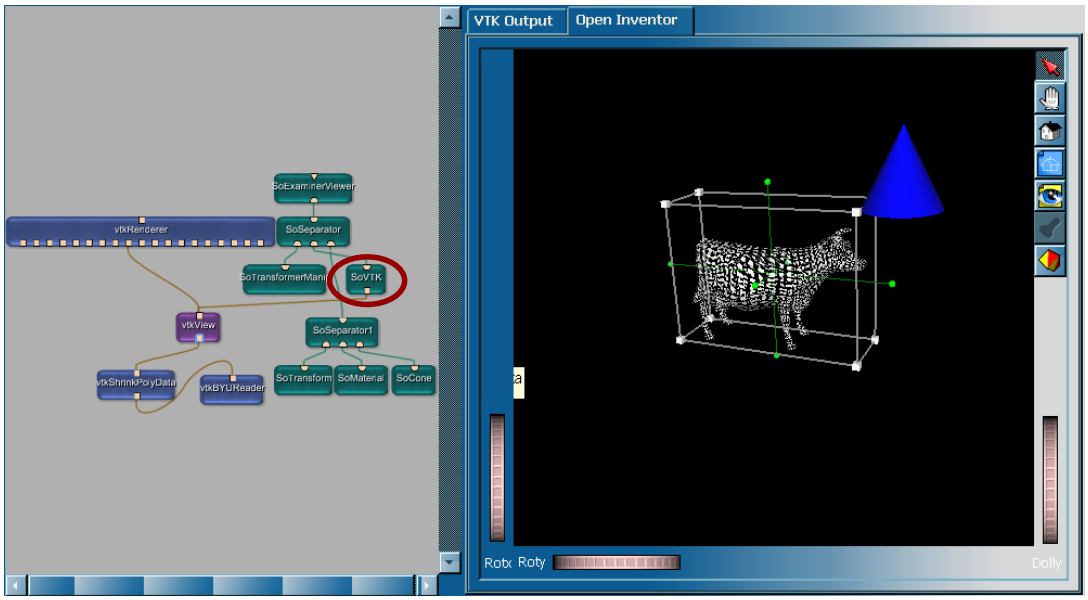
- 2D/3D Image Processing (> 80)
- Grid, Mesh, Surface, and Data Filtering (> 150)
- Pickers (> 5)
- Properties and Actors (> 30)
- Mappers (> 10)
- Renderers, Widgets, Viewers (> 15)
- Sources, Readers and Writers (> 130)
- Transformations (> 10)

VTK Example 1: Contour Filter



VTK Example 2:

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



Automatic wrapper generation

- The ITK and VTK libraries are integrated into MeVisLab using a generic wrapping approach
- This approach facilitates updates to new library versions and makes almost all algorithms of ITK/VTK instantly available
- Other platforms do this wrapping manually and offer a less extensive ITK/VTK integration

MeVisLab SDK

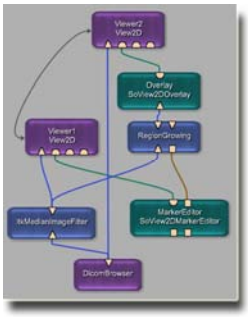
- Allows to extend MeVisLab with
  - ML Modules
  - Open Inventor Modules
  - Macro Modules
- Efficient user interface development
- JavaScript/Python scripting languages

Scripting (MDL)

- User interfaces are created with the „Module Definition Language“ (MDL)
- Abstract hierachical GUI language
- Interpreted at run-time, allows rapid prototyping

www.mevislab.de/fileadmin/docs/html/mdl/index.html

GUI Scripting Example



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
  }
}
```

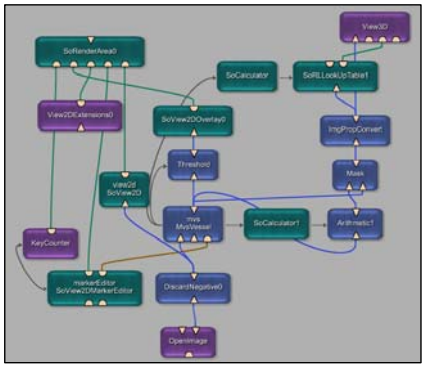
MDL Script

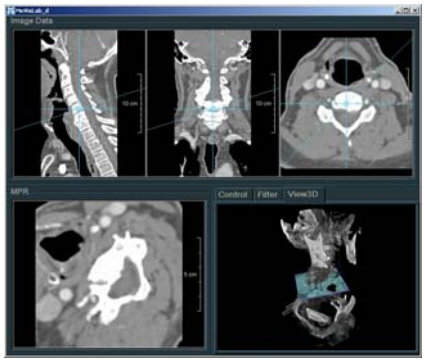
Graphical User Interface

Schematic Representation

Application Prototyping

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





JavaScript / Python Integration

- Scripting can be used to program dynamic behaviour both on network and user interface level
  - Adding modules at run-time
  - Parameter computations and synchronization
  - Dynamic user interfaces
  - External processes
- JavaScript / Python bindings are available

www.mevislab.de/fileadmin/docs/html/script/index.html

Summary

- **MeVisLab Basic** allows to learn about Medical Imaging and Visualization without C++ knowledge
- **Visual Programming** allows easy exploration of algorithms
- OpenInventor, ITK and VTK integrations offer a vast amount of available modules

Get your copy of MeVisLab Basic at:

www.mevislab.de

Acknowledgements

I would like to thanks my colleagues at MeVis for their contributions to this presentation:

T. Boskamp, J. Rexilius, O. Konrad,

W. Spindler and F. Ritter