Outline

MONDAY

VMTK:
Segmentation
Meshing (Fluid and Structure)

What this is about:
Show some tools/capabilities through examples
Show some solutions I have found
for issues I had to face!

TUESDAY

GMSH:
Remeshing, Boundary Flags

What this is NOT about:
Theory about segmenting/meshing
Programming in VMTK/VTK

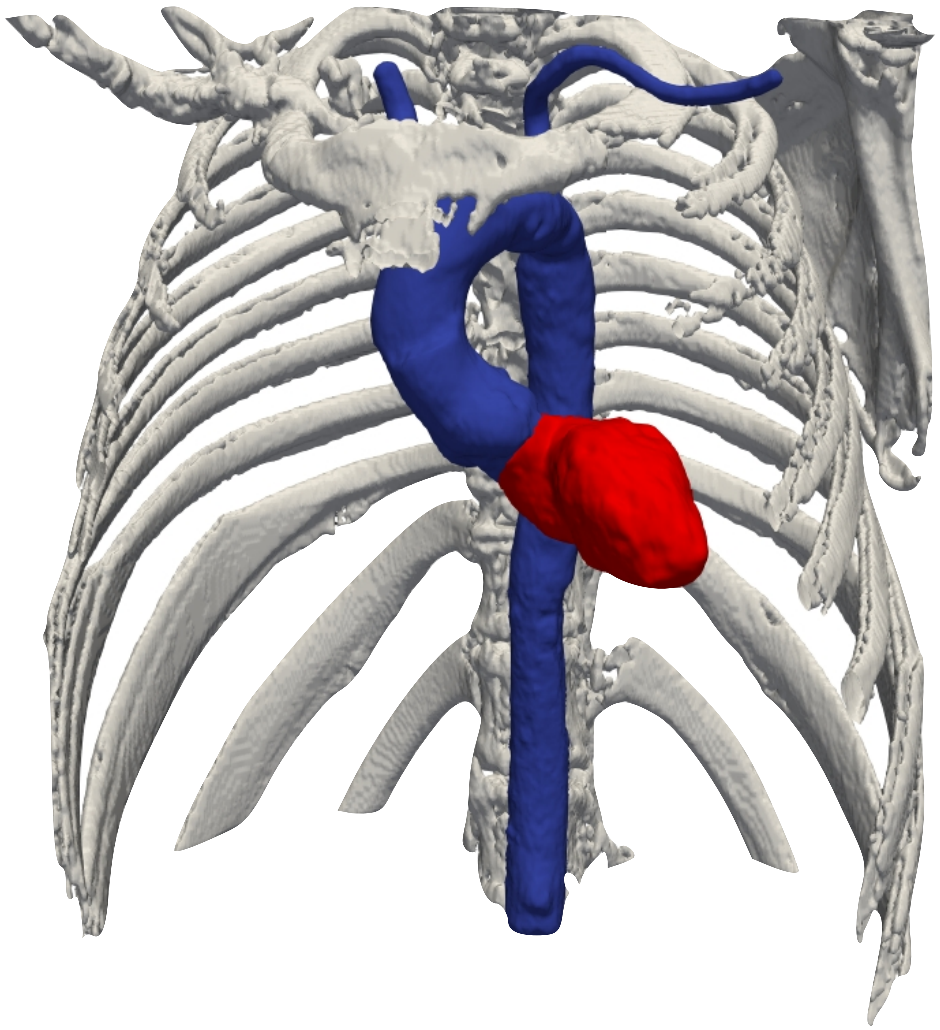
PARAVIEW
Pre Processing
Post Processing

Segmentation

Input data
Medical images

Tool
VMTK
(www.vmtk.org)

Output data
Mesh (surface and volume)



Useful references

- VMTK tutorial webpage
- 3dImageToolkit, GitHub
- E. Faggiano, M. Fedele

VMTK

http://www.vmtk.org/license.html

VMTK LICENSE

VMTK, the Vascular Modeling Toolkit.

VMTK

http://www.vmtk.org/license.html

##APPLICABLE LICENSES

- vmtk is licensed under the BSD license.
- ITK is licensed under the Apache 2.0 license.
- VTK is licensed under the Visualization Toolkit (VTK) license.
- OpenNL is license under a modified BSD license.
- Tetgen is licensed under a modified MIT license. Tetgen is not a required dependency.
- Stellar and Starbase are licensed under the BSD license.

See Copyright for details.

VMTK

HOW TO CITE:

http://vmtk-users.narkive.com/ZnD6ZBsh/vmtk-reference

Usually there are two ways:
- the website: The Vascular Modeling Toolkit website, www.vmtk.org, last accessed 20 Dec 2012
- this paper: Antiga L, Piccinelli M, Botti L, Ene-Iordache B, Remuzzi A and Steinman DA. An image-based modeling framework for patient-specific computational hemodynamics. Medical and Biological Engineering and Computing, 46: 1097-1112, Nov 2008.

IN CASE OF QUESTIONS:

http://www.vmtk.org/community/

Mailing Lists

Mailing list is the best way to get in contact with the vmtk community. It serve as primary support channel where users can help each other to learn how to use vmtk.

The main forum for questions and discussions about the vascular modeling toolkit is hosted on google groups:

vmtk-mailing-list

Segmentation

Since it is very well done…we follow the **vmtk tutorial** online!
http://www.vmtk.org/documentation/getting-started.html

File formats

Images

- **dcm (DICOM Image, only Read)**: Digital Imaging and Communications in Medicine (DICOM) is a standard for handling, storing, printing, and transmitting information in medical imaging. It includes a file format definition and a network communications protocol
- **mha/mhd (Meta Image)**: it specifies a standardized way of expressing the meta- information in the header about the pixel data; .mha: header embedded, .mhd+.raw (or other pixel data formats): header in separate file
- **nrrd (Nearly Raw Raster Data)**: similar to Meta Image, somewhat richer header
- **Analyze (Mayo Clinic Analyze 7.5)**: some issues with handling and reconstructing orientations; superseded by NIfTI; .hdr+.img
- **NIfTI (Neuroimaging Informatics Technology Initiative)**: descendant of the Analyze format (can be converted by just changing the header); interoperable with Analyze but orientation and time are treated properly; .hdr+.img or .nii
- **vti (VTK XML Image Data)**: the Visualization Toolkit (VTK) format for images; XML syntax with embedded binary data; very flexible; possible to represent multiple scalar/vector/tensor data in the same file; does not support image orientation (although it is possible to include custom data expressing orientation matrices)

NB: ask the medical doctor you work with to anonymise the DICOM images BEFORE he gives them to you!
(The other image formats are anonymous)

Segmentation

Since it is very well done…we follow the **vmtk tutorial** online!
http://www.vmtk.org/documentation/getting-started.html

File formats

Surfaces

- **vtp (VTK XML)**: the Visualization Toolkit (VTK) format for surfaces; XML syntax with embedded binary data; PolyData (.vtp) — Serial vtkPolyData (unstructured)
- **stl (Stereolithography)**: STL (STereoLithography) is a file format native to the stereolithography CAD software created by 3D Systems. STL is also known as Standard Tessellation Language. STL files describe only the surface geometry of a three dimensional object without any representation of color, texture or other common CAD model attributes. The STL format specifies both ASCII and binary representations. Binary files are more common, since they are more compact
- **tec (Tecplot)**: TEC files are used by the TECPLOT program, which is a visualization program for technical data.

Segmentation

Since it is very well done…we follow the **vmtk tutorial** online!
http://www.vmtk.org/documentation/getting-started.html

File formats

Meshes

- **vtu (VTK XML)**: the Visualization Toolkit (VTK) format for meshes; XML syntax with embedded binary data; UnstructuredGrid (.vtu) — Serial vtkUnstructuredGrid (unstructured)
- **pvtu (VTK XMLP)**: the Visualization Toolkit (VTK) parallel format for meshes; XML syntax with embedded binary data
- **xda (libMesh, only Write)**: libMesh mesh files consist of two sections, the header and the data. The header contains important size information. It defines the number of elements, number of nodes, etc… that are in the mesh. The data section contains the actual element connectivity, the nodal coordinates, and any boundary condition information
- **msh (Fluent, only Write)**: Fluent mesh files format
- **lifev (LifeV)**: LifeV mesh files format **Extension for LifeV: .mesh**
- **vtk**: the Visualization Toolkit (VTK) format for meshes; ASCII or binary format
- **gambit**: Gambit mesh files format
- **tec (Tecplot)**: TEC files are used by the TECPLOT program, which is a visualization program for technical data.
- **fdneut (FIDAP)**: Fidap LifeV files format
- **tetgen**: Tetgen mesh files format
- **ngneut**: Netgen neutral file format, which is particularly easy to read with a home-made program
- **xml (Dolfyn, only Write)**: XML file for the Dolfyn CFD software

Segmentation

EXAMPLE 1:
ABDOMINAL AORTA (HEALTHY)

Go to:
http://www.vmtk.org/tutorials/

Download the file:
Aorta_voi.zip
(End of page, Tutorials Data section)

Segmentation

Since it is very well done…we follow the **vmtk tutorial** online!
http://www.vmtk.org/tutorials/ImageBasedModeling.html

Level Sets

There's a huge amount of literature concerning level sets, so refer to that if you don't know what they are. For now we'll just say that level sets are a kind of deformable model in which the deformable surface is not represented by a set of points and triangles, but rather described by a 3D function (basically another image) whose contour at level zero is the surface in question. In practice, the output of a level set segmentation is an image. To extract the surface from the image you have to run the output image through Marching Cubes with level 0. The advantage of using a deformable model (either explicit or level sets) is that the location of the surface will not depend on the level you choose, but instead it will locally conform to the peaks of the gradient modulus of the image levels. In other words, the final surface will be located on the regions corresponding to the steepest change of image intensity across the vessel wall, which is a robust and objective criterion.

In order to use level sets in vmtk you have to use *vmtklevelsetsegmentation:*

```
vmtklevelsetsegmentation -ifile image_volume_voi.vti -ofile level_sets.vti
```

Recall that *-ifile* and *-ofile* are ways to access the built-in readers and writers that every script has. The file format is guessed from the filename extension.

Depending on the installation, you may have to run the script with:

```
vmtk vmtklevelsetsegmentation -ifile image_volume_voi.vti -ofile level_sets.vti
```

Segmentation

Since it is very well done…we follow the **vmtk tutorial** online!
http://www.vmtk.org/tutorials/ImageBasedModeling.html

A message will appear:

```
1.  Please choose initialization type: (0: colliding fronts; 1: fast marching; 2: threshold; 3
    : isosurface)
```

This lets you choose the way in which your deformable model is initialized. The goal is to initialize the model as close to the vessel wall as possible.

Colliding Fronts: This initialization type consists of placing two seeds on the image. Two fronts will be propagated from the seeds (one front from each) with their speeds proportional to the image intensity. The region where the two fronts cross,(or collide) is then the initial deformable model. This type of initialization is very effective when it is necessary to initialize the tract of a vessel. For example, by placing a seed at each of the two extremities (thus, two seeds total), the region between the seeds will then be selected. An advantage to this method is that side branches will be ignored.

Fast Marching: This initialization type consists of placing a set of seeds and a set of targets on the image. A front is then propagated from the seeds until the first target is met at which point the region covered by the front is the initial deformable model. This type of initialization is effective when it is necessary to segment round objects such as aneurysms. For example, by simply placing one seed at the center and one target on the wall, the volume will be initialized.

Threshold: With this initialization type, pixels comprised within two specified thresholds will be selected as the initial level sets.

Isosurface: With this initialization type, initial level sets will correspond to an isusurface of the image with sub-pixel precision.

Segmentation

Since it is very well done…we follow the **vmtk tutorial** online!
http://www.vmtk.org/tutorials/ImageBasedModeling.html

Enter *0* to initialize with colliding fronts.

A message will then appear:

```
1.  Please input lower threshold ('n' for none):
```

Wave propagation can be restricted to a set of intensity levels above a lower threshold and below an upper threshold. This is useful when you don't have a great SNR* in you images. The use of such thresholds is optional, and in general they do not influence the location of the final surface. If you don't know what the right threshold is, press i to activate the image and probe it. Quit with q when probing is done. Next, enter the value returned by the probe on the terminal. If you don't want to use any threshold, just enter n (give this a try before using thresholds).

The next message is:

```
1.  Please input upper threshold ('n' for none):
```

The above procedure applies here as well.

Next, you'll be prompted with:

```
1.  Please place two seeds (click on the image while pressing Ctrl).
```

**SNR=Signal-to-Noise Ratio*

Segmentation

Since it is very well done…we follow the **vmtk tutorial** online!
http://www.vmtk.org/tutorials/ImageBasedModeling.html

The following message will now appear:

```
Please input parameters (type return to accept current values, 'q' to quit):
NumberOfIterations(0) [PropagationScaling(1.0) CurvatureScaling(0.0) AdvectionScaling(0.0)
]:
```

These parameters control the deformation of your level set.

Number of iterations is the number of deformation steps the model will perform. For numerical reasons, the distance the model will travel will depend on a number of things, among which are voxel size and image gradient modulus intensity.

Propagation scaling is the weight you assign to model inflation.

Curvature scaling is the weight you assign to model surface regularization (this will eventually make the model collapse and vanish if it’s too strong)

Advection scaling is the most important weight. It regulates the attraction of the surface of the image gradient modulus ridges, which is ultimately what you want.

From experience it is recommended that propagation and curvature should be set to *0.0*, and advection to *1.0*. This is possible if the initialization is sufficiently close to where you want the surface to converge. The number of iterations should be set large enough for the level set not to move anymore (if the region isn’t too big, try with *300*). Experiment with the images to see what happens. Keep in mind that setting advection to *1.0* and propagation and curvature to *0.0* will robustly lead you to reproducible results.

Segmentation

Since it is very well done…we follow the **vmtk tutorial** online!
http://www.vmtk.org/tutorials/ImageBasedModeling.html

At this point you have a file named level_sets.vti which contains an image (which you can display for examination using vmtkimageviewer).

```
(vmtk) vmtkimageviewer -ifile level_set.vti
```

The zero level of this image is the surface you generated. Therefore, now extract a polygonal surface from it:

```
1.  vmtkmarchingcubes -ifile level_sets.vti -ofile model.vtp
```

To get information about a script:

```
(vmtk) vmtkmarchingcubes —help
```

To visualise the resulting surface mesh:

```
(vmtk) vmtksurfaceviewer  -ifile model.vtp
```

Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 0:
Extract **volume of interest**

Volume of interest (VOI) extraction

Often, medical images contain structures which are not of interest. To extract a volume of interest (VOI) from a dataset, one can pipe an instance of vmtkimagevoiselector between the DICOM reader and the writer in this way:

```
(vmtk) vmtkimagevoiselector -ifile image_volume.vti -ofile image_volume_voi.vti
```

When the render window pops up, pressing i will activate the interactor. A yellowish cube will then appear which will be the tool used to select the VOI.

- **Translate the cube** by middle-clicking on it.
- **Resize the cube** by left-clicking and dragging the little spheres (handles) on the faces of the cube.
- **Normal interaction** with the image planes is still active, so you can still move them as explained before.

When satisfied with the VOI, press q.

http://www.vmtk.org/documentation/getting-started.html

Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 1:
Construct the **level set function** of the **lumen** of the abdominal aorta + iliac arteries

```
(vmtk) vmtklevelsetsegmentation -ifile image_volume_voi.vti -ofile level_set_lumen.vti
```

Colliding Fronts: This initialization type consists of placing two seeds on the image. Two fronts will be propagated from the seeds (one front from each) with their speeds proportional to the image intensity. The region where the two fronts cross,(or collide) is then the initial deformable model. This type of initialization is very effective when it is necessary to initialize the tract of a vessel. For example, by placing a seed at each of the two extremities (thus, two seeds total), the region between the seeds will then be selected. An advantage to this method is that side branches will be ignored.

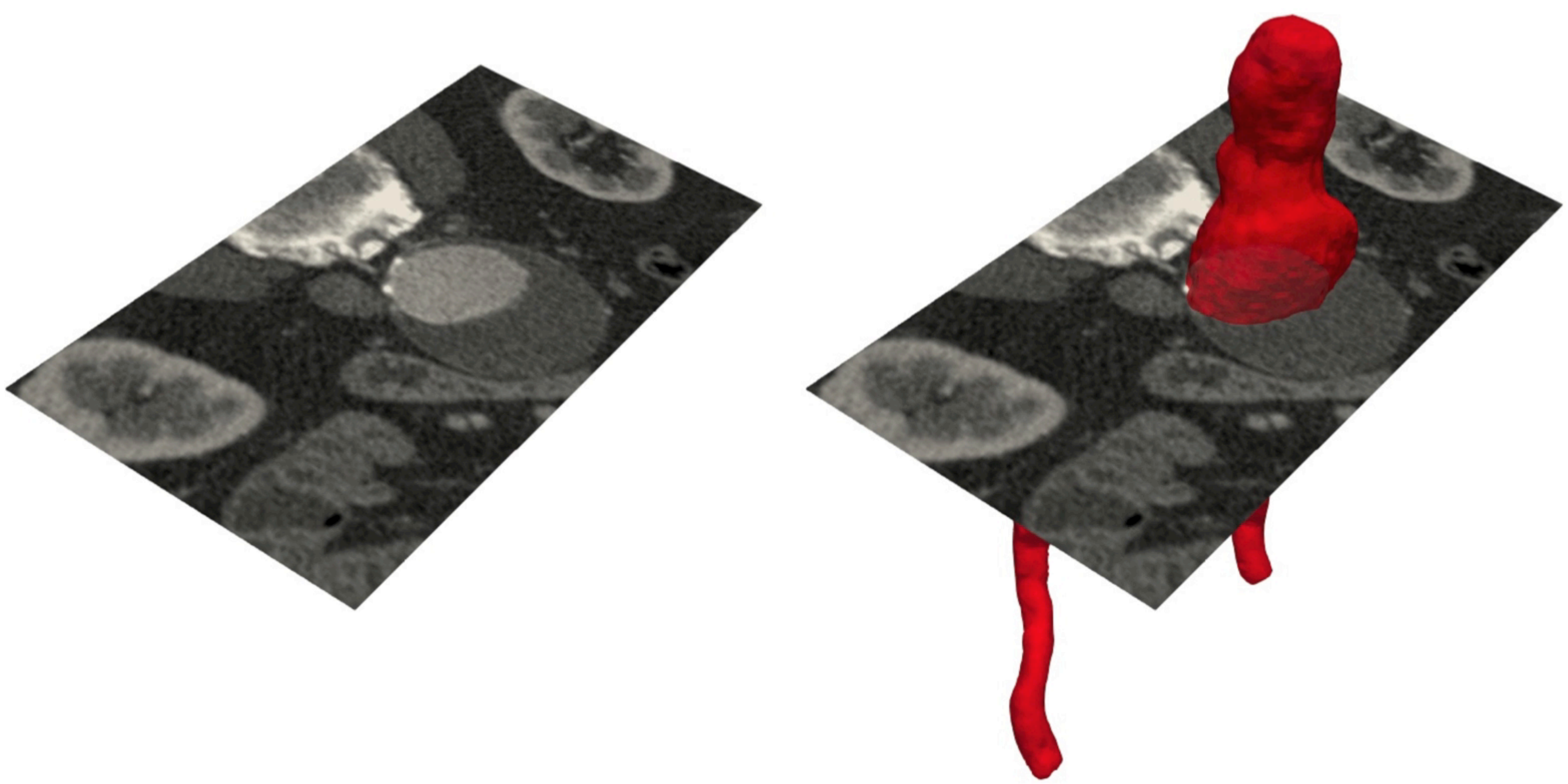
http://www.vmtk.org/tutorials/ImageBasedModeling.html

Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 1:
Construct the **level set function** of the **lumen** of the abdominal aorta + iliac arteries

```
(vmtk) vmtklevelsetsegmentation -ifile image_volume_voi.vti -ofile level_set_lumen.vti
```



http://www.vmtk.org/tutorials/ImageBasedModeling.html

Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 2:
Construct the **level set function** of the **aneurysm sac**

```
(vmtk) vmtklevelsetsegmentation -ifile image_volume_voi.vti -ofile level_set_aaa.vti
```

Fast Marching: This initialization type consists of placing a set of seeds and a set of targets on the image. A front is then propagated from the seeds until the first target is met at which point the region covered by the front is the initial deformable model. This type of initialization is effective when it is necessary to segment round objects such as aneurysms. For example, by simply placing one seed at the center and one target on the wall, the volume will be initialized.

http://www.vmtk.org/tutorials/ImageBasedModeling.html

Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 3:

Construct the **level set function** of the abdominal aortic **aneurysm + neck + iliac**

This result can be achieved combining together the level set images obtained in **task1** e **task2**. Since the level set image has negative value inside the surface and positive value outside, to obtain the level set of the entire aneurysm we should take the **"min"** of the two images.

vmtkimagecompose

Description

compose an image based on user-specified parameters or on a reference image

```
(vmtk) vmtkimagecompose -ifile level_set_lumen.vti -i2file level_set_sac.vti
-ofile level_set_aaa.vti
```

http://www.vmtk.org/vmtkscripts/vmtkimagecompose.html

Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

Input arguments

Argument	Variable	Type	Length	Range	Default	Description
id	Id	str	1		0	script id
handle	Self	self	1			handle to self
disabled	Disabled	bool	1		0	disable execution and piping
i	Image	vtkImageData	1			the input image
ifile	ImageInputFileName	str	1			filename for the default Image reader
i2	Image2	vtkImageData	1			the second input image
i2file	Image2InputFileName	str	1			filename for the default Image2 reader
operation	Operation	str	1	["min","max","multiply","subtract"]	min	the operation used to compose images
negatei2	NegateImage2	bool	1		False	negate the second input before composing
ofile	ImageOutputFileName	str	1			filename for the default Image writer

http://www.vmtk.org/vmtkscripts/vmtkimagecompose.html

Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 3:

Construct the **level set function** of the abdominal aortic **aneurysm + neck + iliac**

This result can be achieved combining together the level set images obtained in **task1** e **task2**. Since the level set image has negative value inside the surface and positive value outside, to obtain the level set of the entire aneurysm we should take the **"min"** of the two images.

vmtkimagecompose

Description

compose an image based on user-specified parameters or on a reference image

```
(vmtk) vmtkimagecompose -ifile level_set_aaa.vti -i2file level_set_sac.vti
-ofile level_set_ilt.vti –pipe vmtkimageviewer -ifile level_set_aaa.vti
```

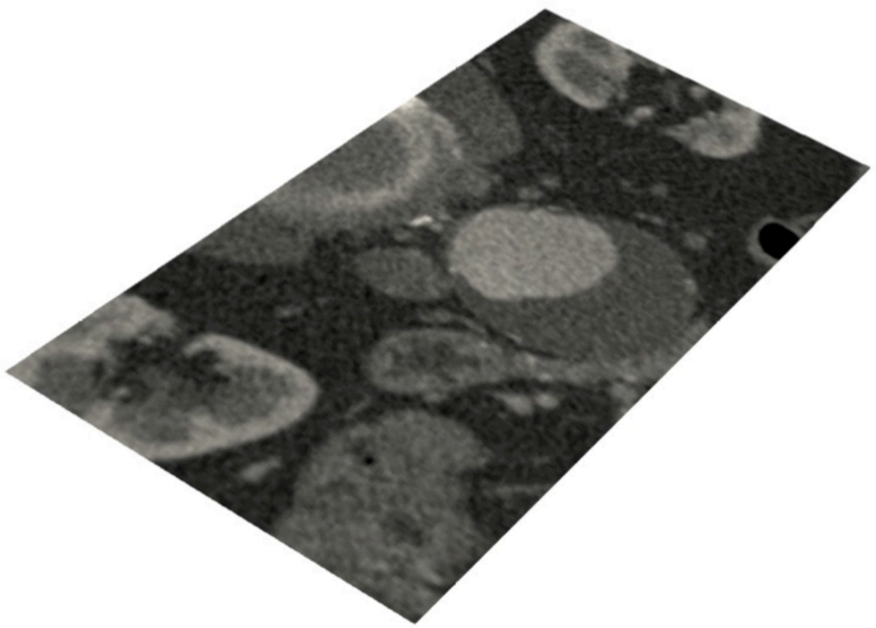
http://www.vmtk.org/vmtkscripts/vmtkimagecompose.html

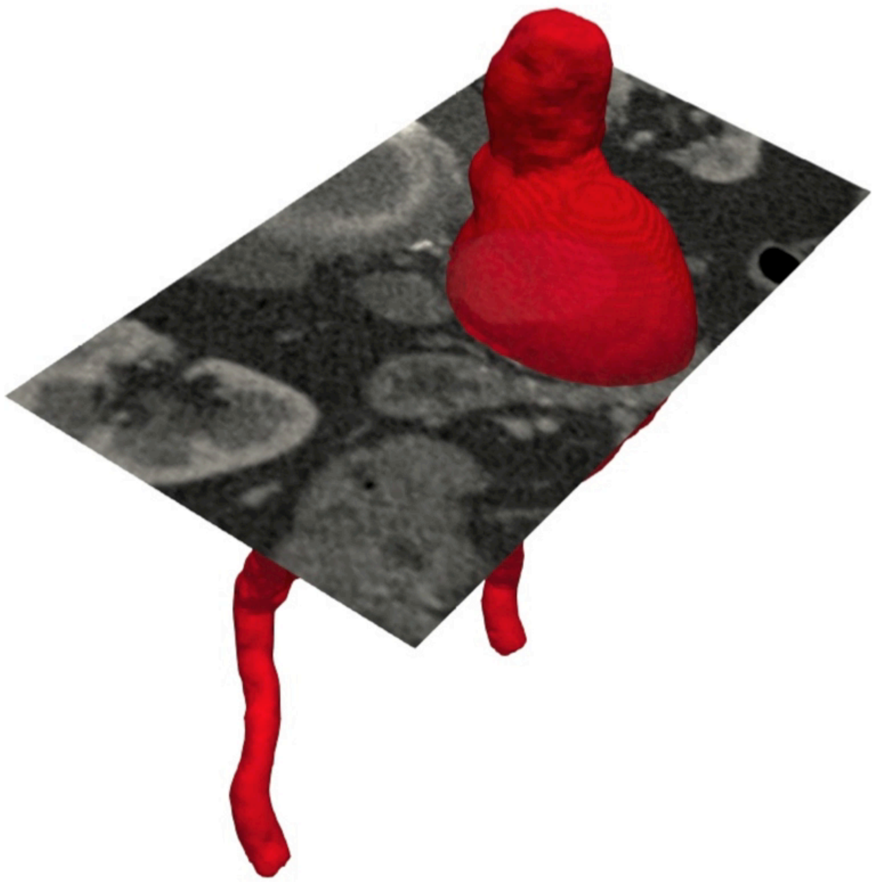
Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 3:
Construct the **level set function** of the abdominal aortic **aneurysm + neck + iliac**

```
(vmtk) vmtkimagecompose -ifile level_set_aaa.vti -i2file level_set_sac.vti
-ofile level_set_ilt.vti  -pipe vmtkimageviewer -ifile level_set_aaa.vti
```





Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 4:
Construct the **level set function** of the **intraluminal thrombus**

This result can be achieved combining together the level set images obtained in **task1** e **task2**.
First we cut the levelsetimages such that they contain only the abdominal aorta.

```
(vmtk) vmtkimageviewer -ifile level_set_sac.vti
```

Probe the box bounds coordinates of the aneurysm sac and cut the image.

```
(vmtk) vmtkimagevoiselector -ifile level_set_sac.vti -ofile
level_set_sac_cut.vti
```

Cut also the level set image of the lumen:

```
(vmtk) vmtkimagevoiselector -ifile level_set_lumen.vti -ofile
level_set_lumen_cut.vti
```

Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 4:
Construct the **level set function** of the **intraluminal thrombus**

This result can be achieved combining together the level set images obtained in **task1** e **task2**.
First we cut the levelsetimages such that they contain only the abdominal aorta.

Since the level set image has negative value inside the surface and positive value outside, to obtain the level set of the intraluminal thrombus we should **"multiply"** the two images.

```
(vmtk) vmtkimagecompose -ifile level_set_lumen_cut.vti -i2file
level_set_sac_cut.vti -ofile level_set_ilt.vti  -operation “multiply“ —pipe
vmtkimageviewer -ifile level_set_ilt.vti
```

Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 5:
Construct the **surface mesh** for the level set obtained at task1, task3

At this point you have a file named level_sets.vti which contains an image (which you can display for examination using vmtkimageviewer). The zero level of this image is the surface you generated. Therefore, now extract a polygonal surface from it:

```
1.  vmtkmarchingcubes -ifile level_sets.vti -ofile model.vtp
```

```
(vmtk) vmtkmarchingcubes -ifile level_set_lumen.vti -ofile model_lumen.vtp
```

```
(vmtk) vmtkmarchingcubes -ifile level_set_aaa.vti -ofile model_aaa.vtp
```

Aneurysm segmentation

EXAMPLE 2:
ABDOMINAL AORTIC ANEURYSM

TASK 6:
Exercise: add **renal arteries**

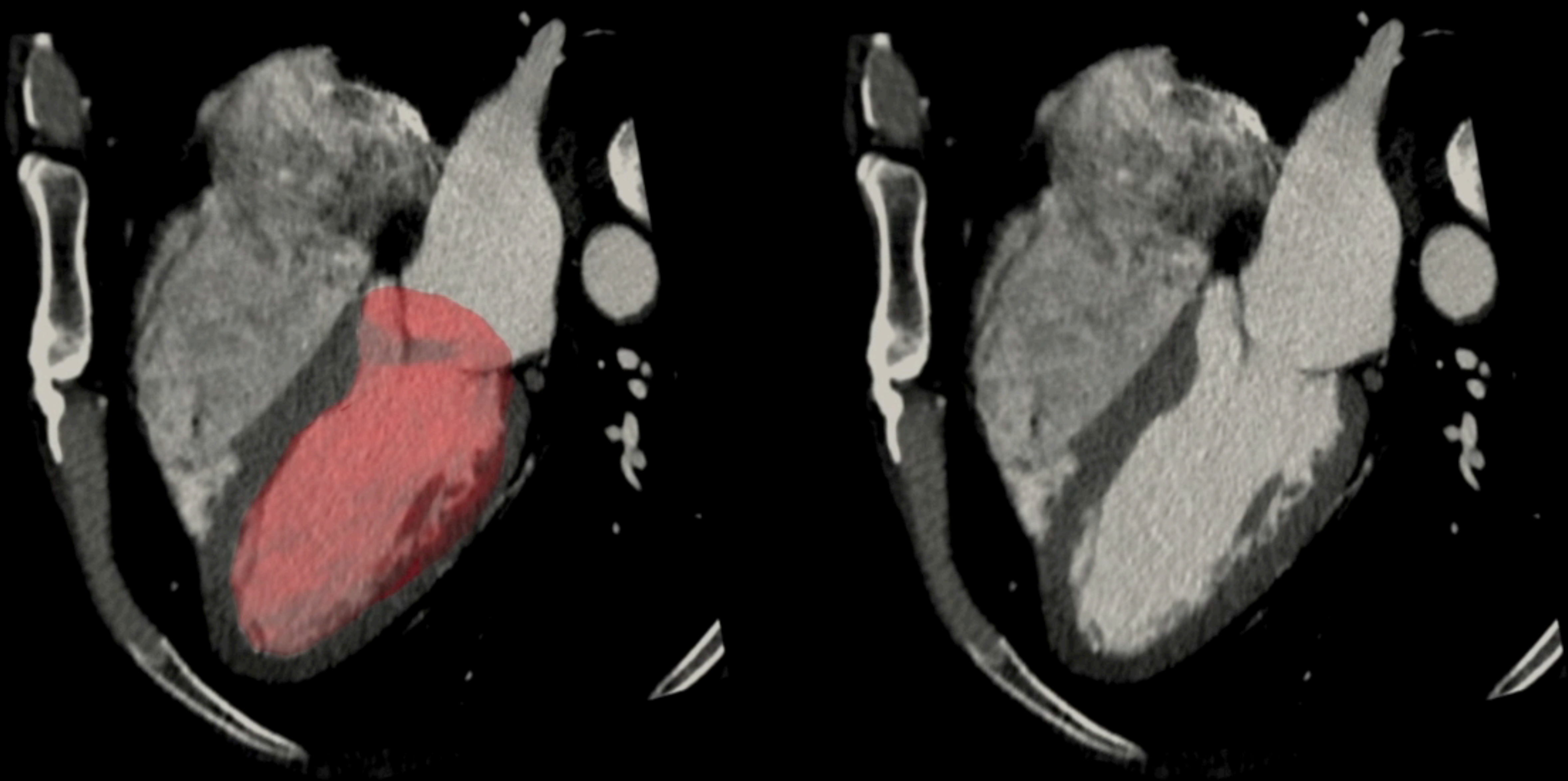
1. Construct the level set of the renal arteries
2. Use vmtkimagecompose to create a level set that includes the aorta lumen plus the renal arteries
3. Construct the surface mesh

TASK 7:
Exercise: construct the model of the **scheleton**

Tip: in vmtklevelsetsegmentation, select "threshold"

Segmentation

Using the same techniques you can reconstruct other parts:



NB: the result depends on the quality of the input images!