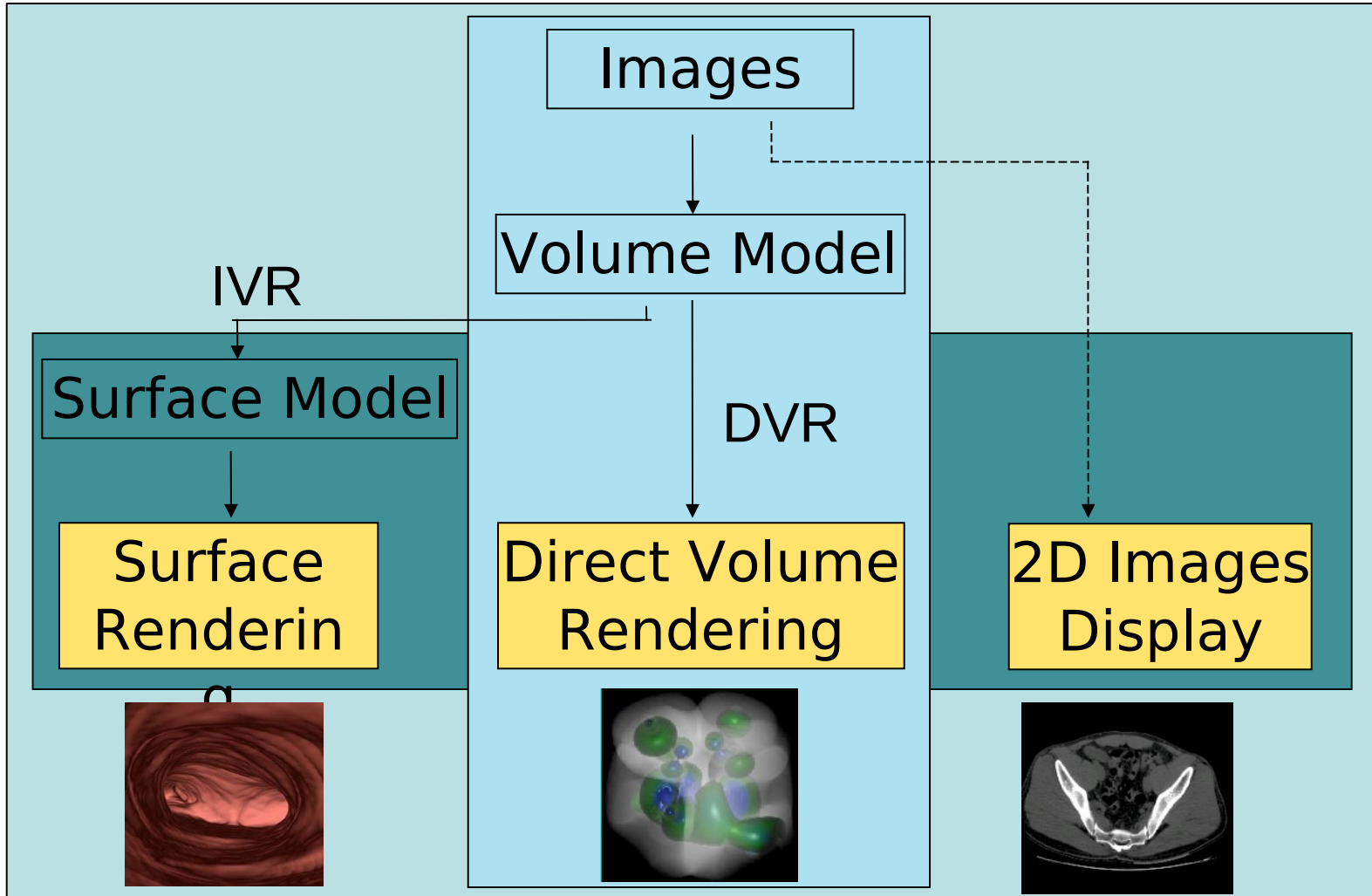


Direct Volume Rendering (II)

Indirect Volume Rendering

D. Tost

Volume rendering

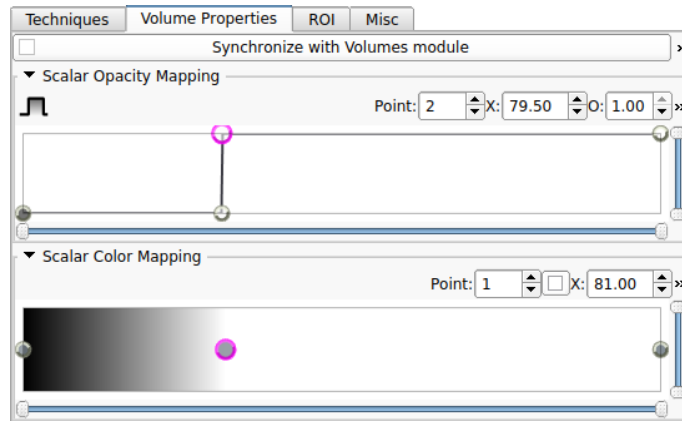


Last session ...

We studied DVR. Let's review it with a couple of exercises ..

What is the intensity of light at a point P located in voxel V of value 130 taking into account the following rendering conditions:

- nearest-neighbor interpolation
- emission + absorption volumetric shading (see TF below)

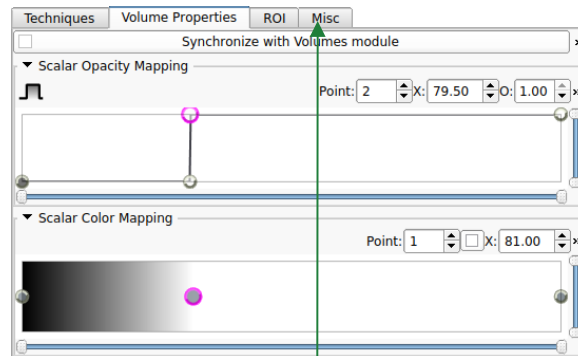


Exercise

Last session, we studied DVR. Let's review it with a couple of exercises ..

What is the intensity of light at a point P located in voxel V of value 130 taking into account the following rendering conditions:

- nearest-neighbor interpolation $\Rightarrow \text{value}(P) = \text{value}(\text{voxel}(P)) = 130$
- emission + absorption volumetric shading (see TF at right)



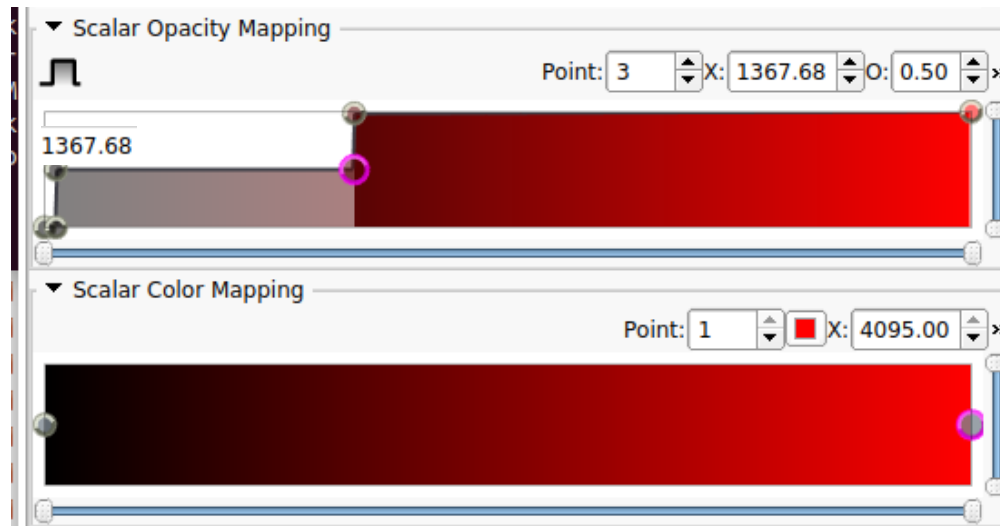
130

$\alpha = 1.0, E = (1.0, 1.0, 1.0)$

Answer: $I(P) = (1.0, 1.0, 1.0)$

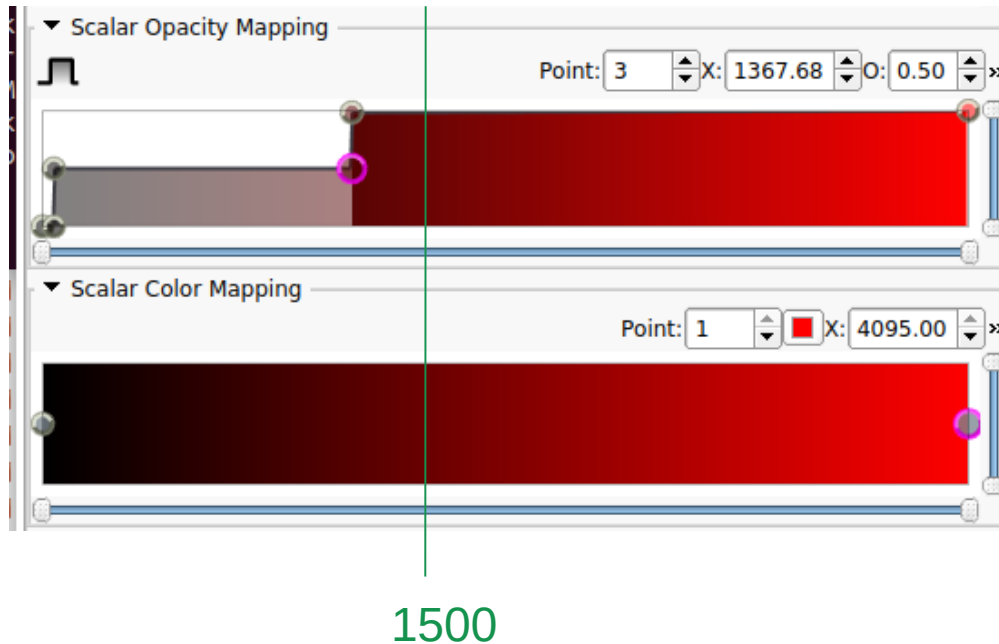
Exercise 2

Same exercise with the following transfer function and voxel value 1500:



Exercise 2

Same exercise with the following transfer function and voxel value 1500:



alpha = 1.0,

G, B = 0
 R interpolated between 0
 and 4095 at 1500

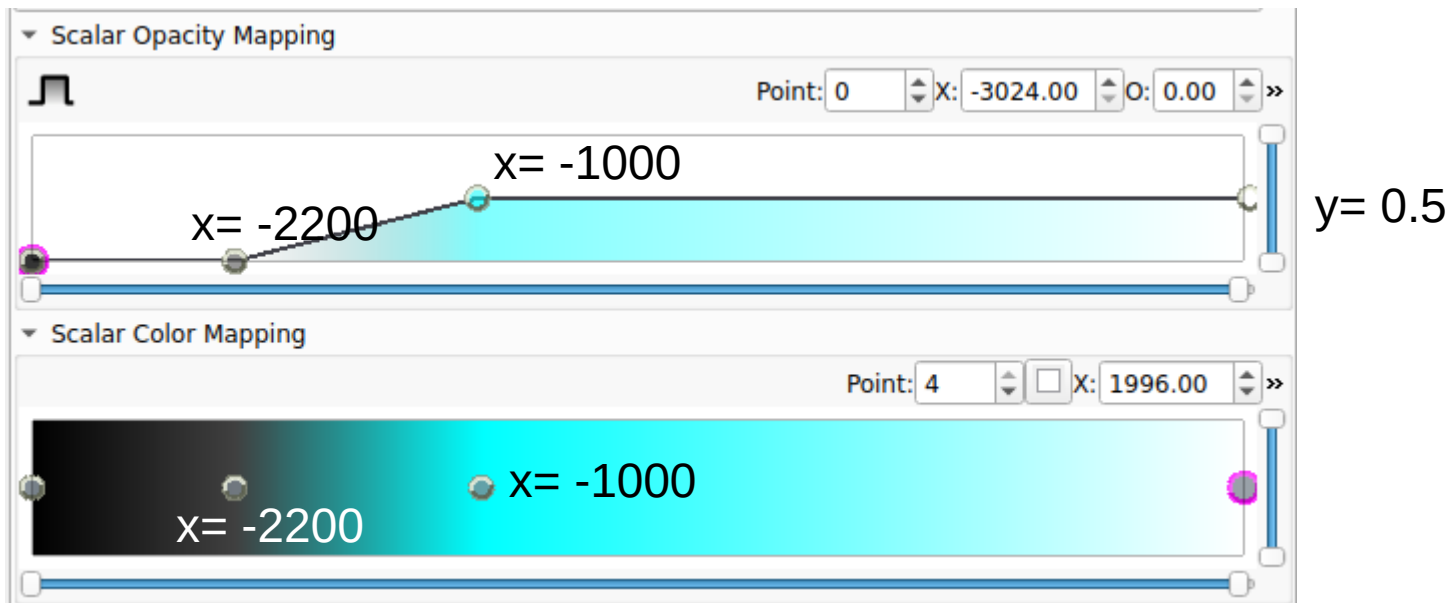
$$R = 1500/4095 = 0.37$$

$$E = (0.37, 0, 0)$$

Answer: $I(P) = (0.37, 0, 0)$

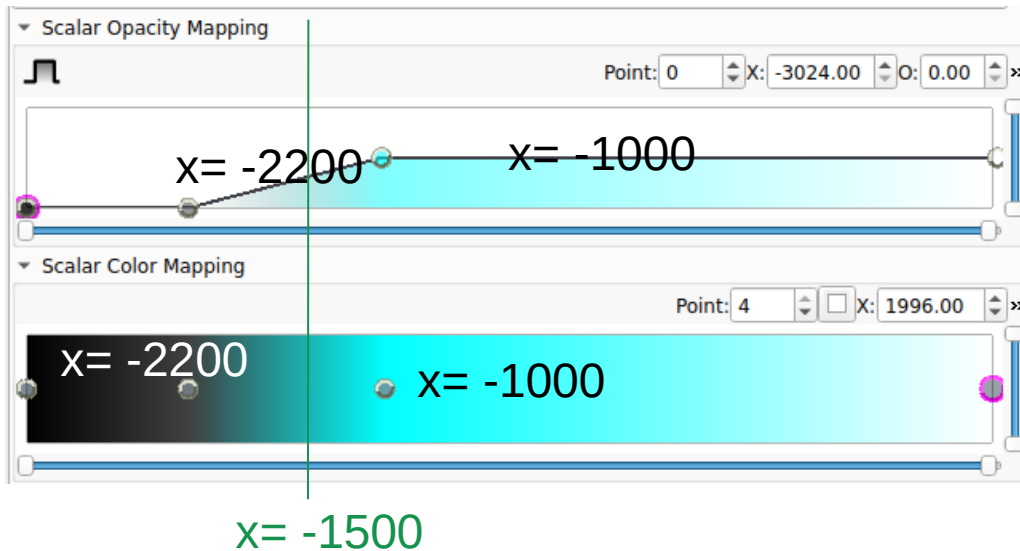
Exercise 3

Same exercise with the following transfer function and voxel value -1500:



Exercise 3

Same exercise with the following transfer function and voxel value -1500:



Answer: $I(P) = (0, 0.17, 0.17)$

$$\text{Alpha: } (A-0) / (-1500 - (-2200)) = (0.5-0) / (-1000 - (-2200))$$

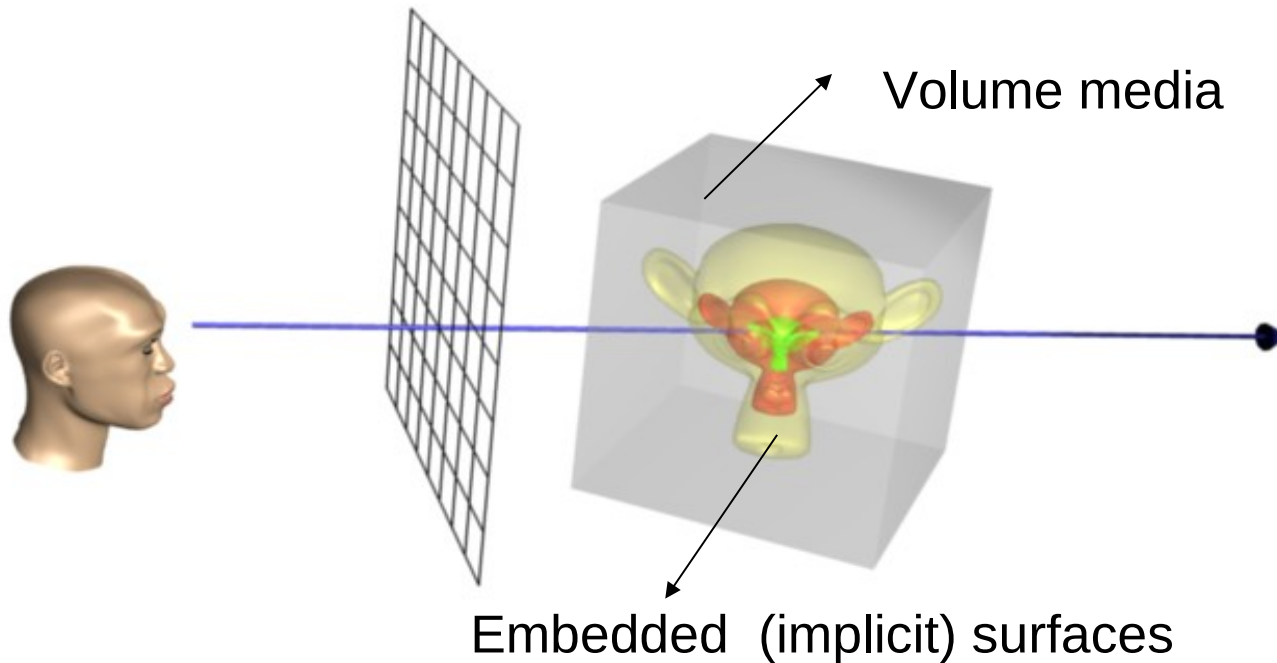
$$A = 0.29$$

$$\text{Emission: } (0, M, M) \text{ where } (M-0) / ((-1500 - (-2200))) = (1-0) / (-1000 - (-2200))$$

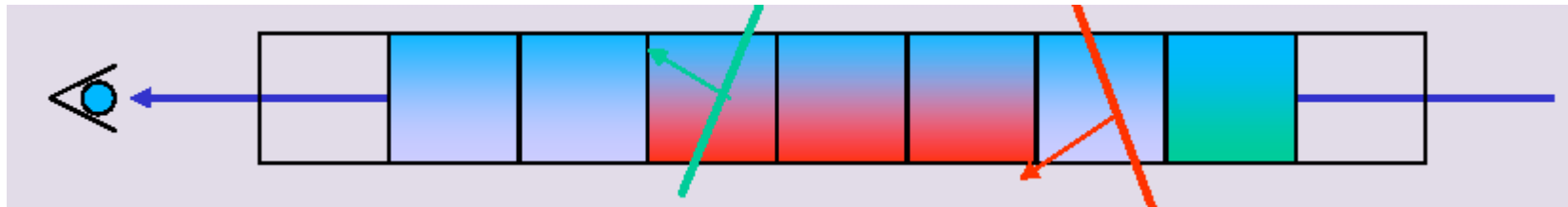
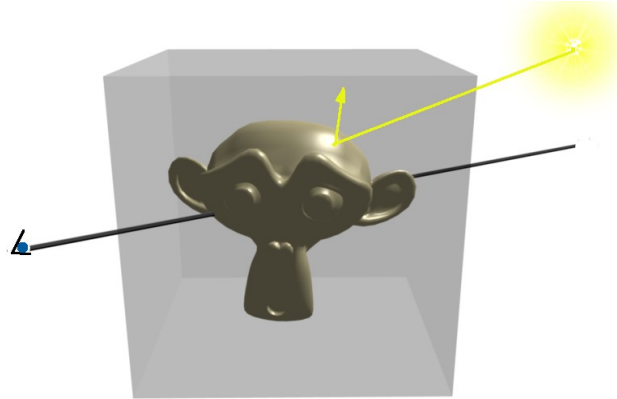
$$(0, 0.58, 0.58)$$

Direct volume rendering (DVR)

Need to compute the transport of light through volume media (volume shading) and on embedded surfaces (surface shading)



Surface + volume shading

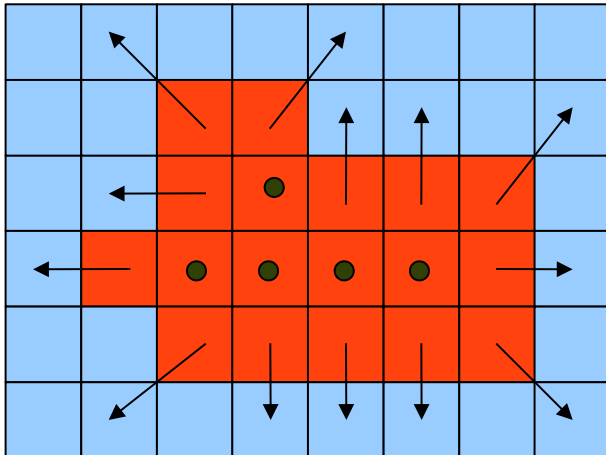


$$I_n = \sum \left(E_i + L_{\text{Surf}_i} \right) \alpha_i l_i \prod \left(1 - \alpha_j l_j \right)$$

$$L_{\text{Surf}_i} = I_a k_a O_d + \sum I_k \left(k_d O_d N \cdot L_k + k_s O_s \left(R_k \cdot V \right)^n \right)$$

Shading

- How to detect surfaces? Searching significant variations of the property value.
- How to compute the normal vector at a sample point? As the normalized gradient vector.



- $N_x = G_x$ (difference in x)
- $N_y = G_y$ (difference in y)
- $N_z = G_z$ (difference in z)

Gradient

Forward difference

- $G_x = V_{(i+1)jk} - V_{ijk}$
- $G_y = V_{i(j+1)k} - V_{ijk}$
- $G_z = V_{ij(k+1)} - V_{ijk}$

Backward difference

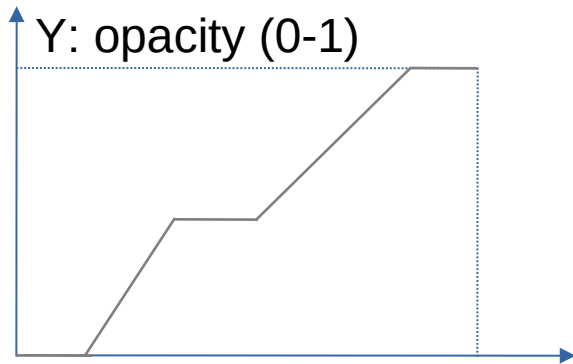
- $G_x = V_{ijk} - V_{(i-1)jk}$
- $G_y = V_{ijk} - V_{i(j-1)k}$
- $G_z = V_{ijk} - V_{ij(k-1)}$

Central difference

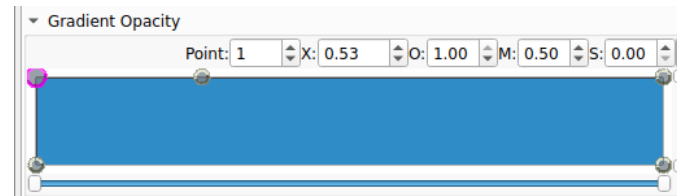
- $G_x = (V_{(i+1)jk} - V_{(i-1)jk}) / 2$
- $G_y = (V_{i(j+1)k} - V_{i(j-1)k}) / 2$
- $G_z = (V_{ij(k+1)} - V_{ij(k-1)}) / 2$

Gradient Opacity Transfer Function

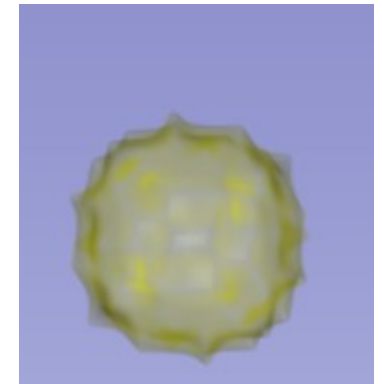
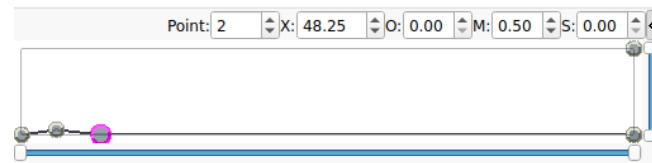
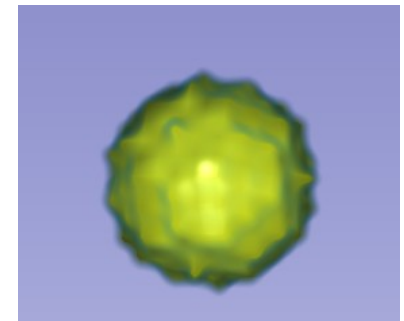
This transfer function defines the opacity in relation to the gradient value. It is used to get special effects of glass-like transparency.



X: gradient norm value
In a range of 0 to 255



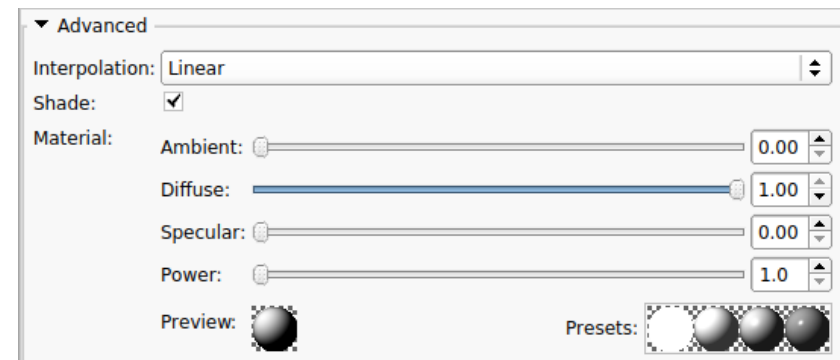
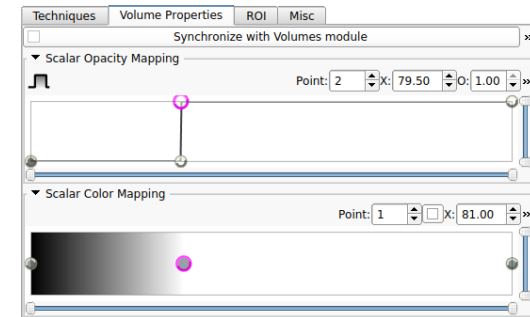
Default value



Exercise 4

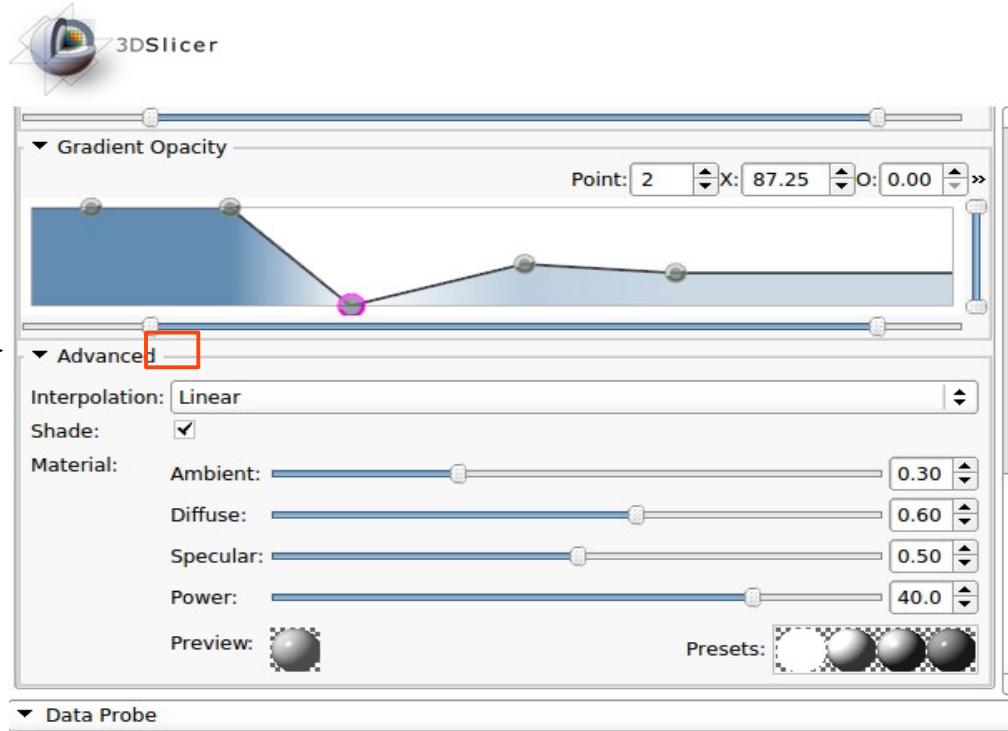
What is the intensity of light at a point P located in voxel V of value 130 and gradient vector $G=(1, 0, 2)$ taking into account the following rendering conditions:

- nearest-neighbor interpolation
- minimum significant gradient: $|G| > 1$
- emission + absorption volumetric shading (see TF at right)
- diffuse only surface shading (see props at right)
- a unique direction light source:
 - vector $L_F = (1.0, 0.0, 0.0)$
 - Intensity $I_F = (1.0, 1.0, 1.0)$
 - Viewing vector $V = (0.86, 0.5, 0)$




Practice

Turn on/off surface shading and play



Put it on →

Practice with the optical properties.
 $O_d = E$
 $O_s = (1, 1, 1)$

Slice Annotations: 

Red RAS: (-113.3, 90.3, -3.3) Axial Sp: 0.9

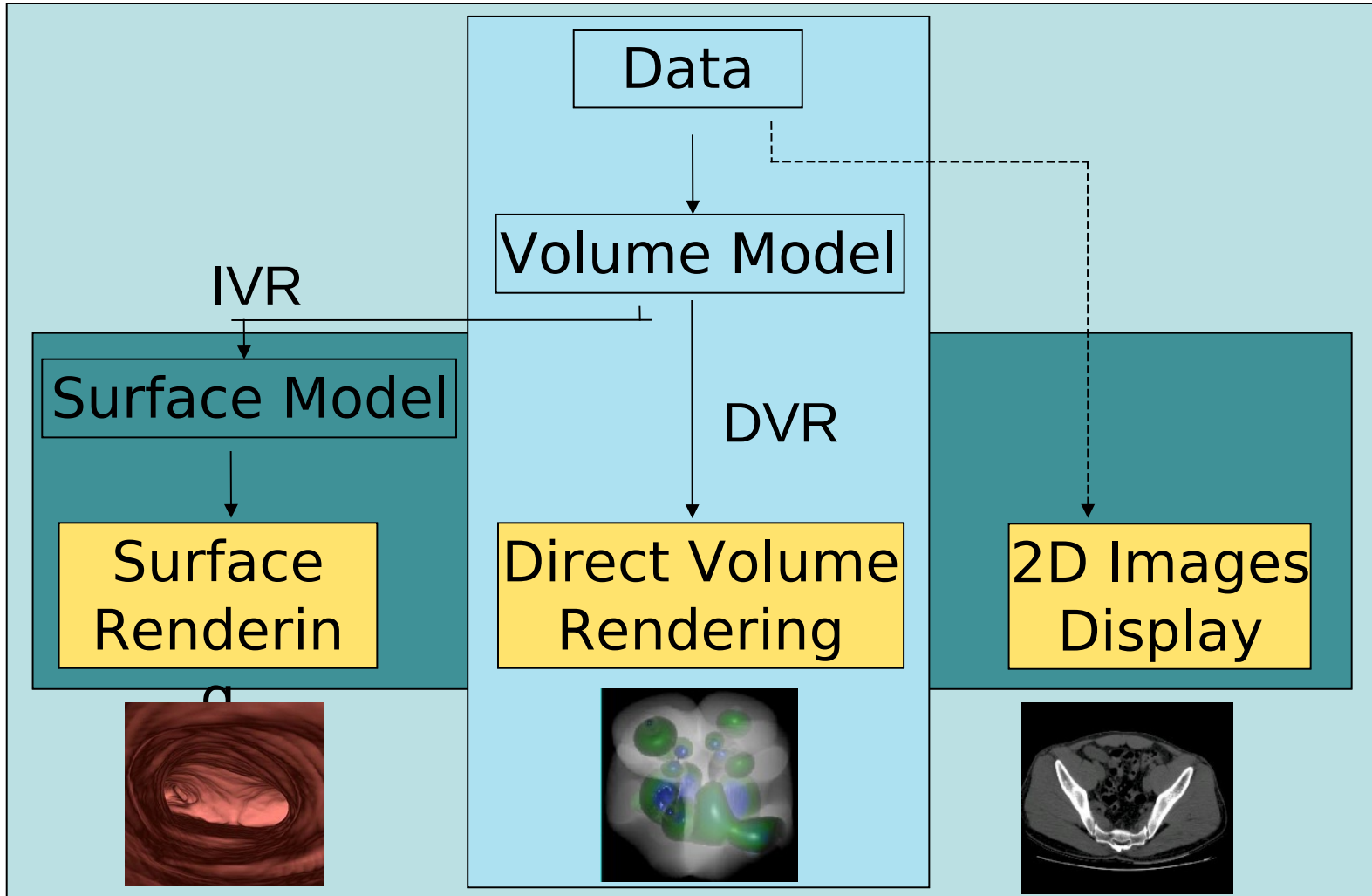
L None

F None

B grayscale (248, 124, 140) 0

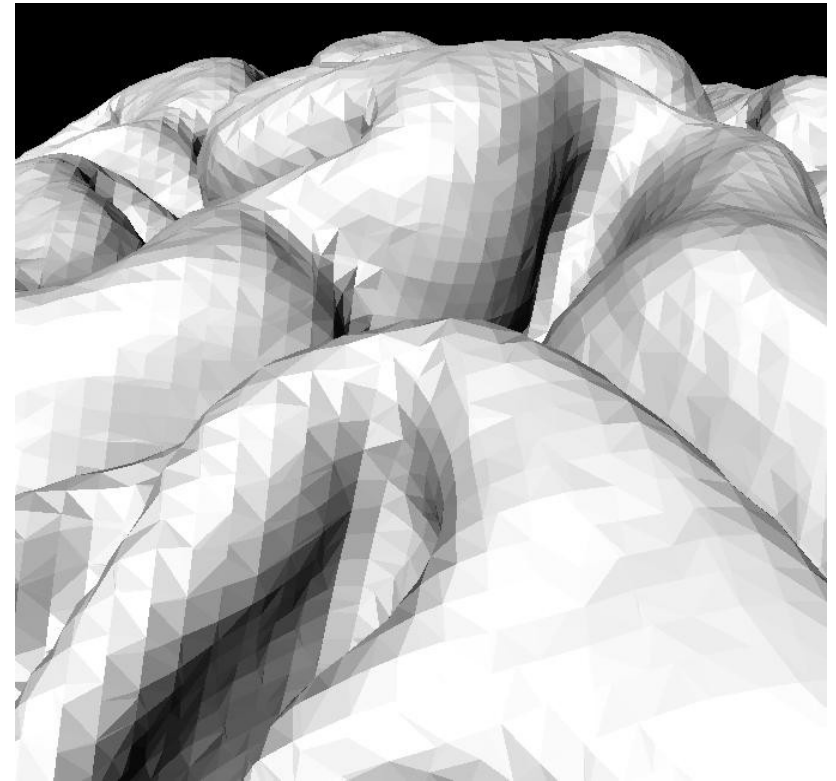
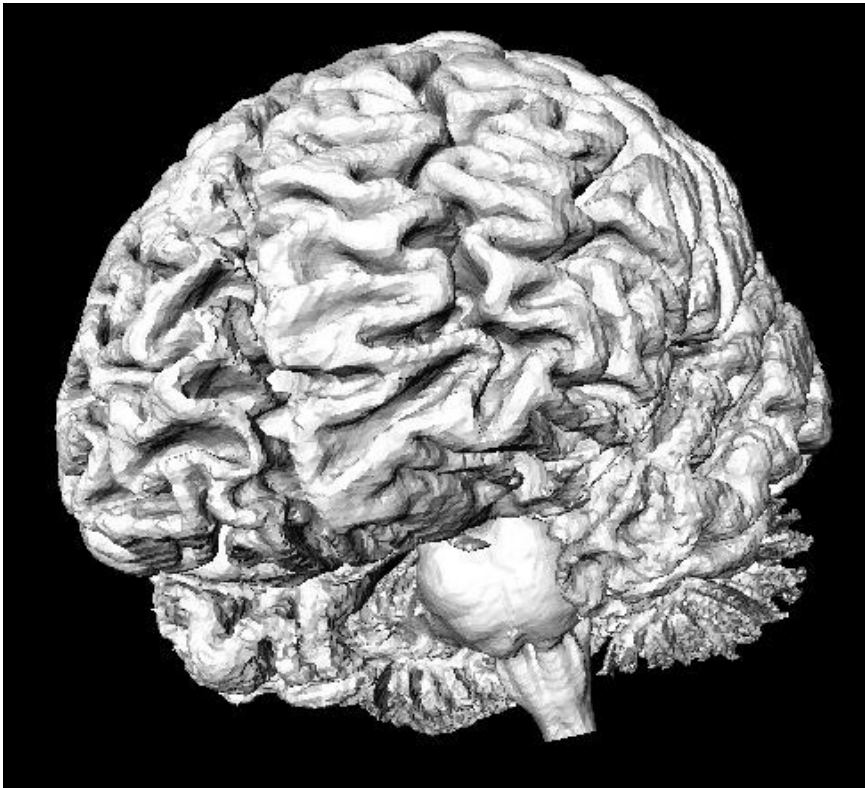
Indirect volume rendering

Volume rendering



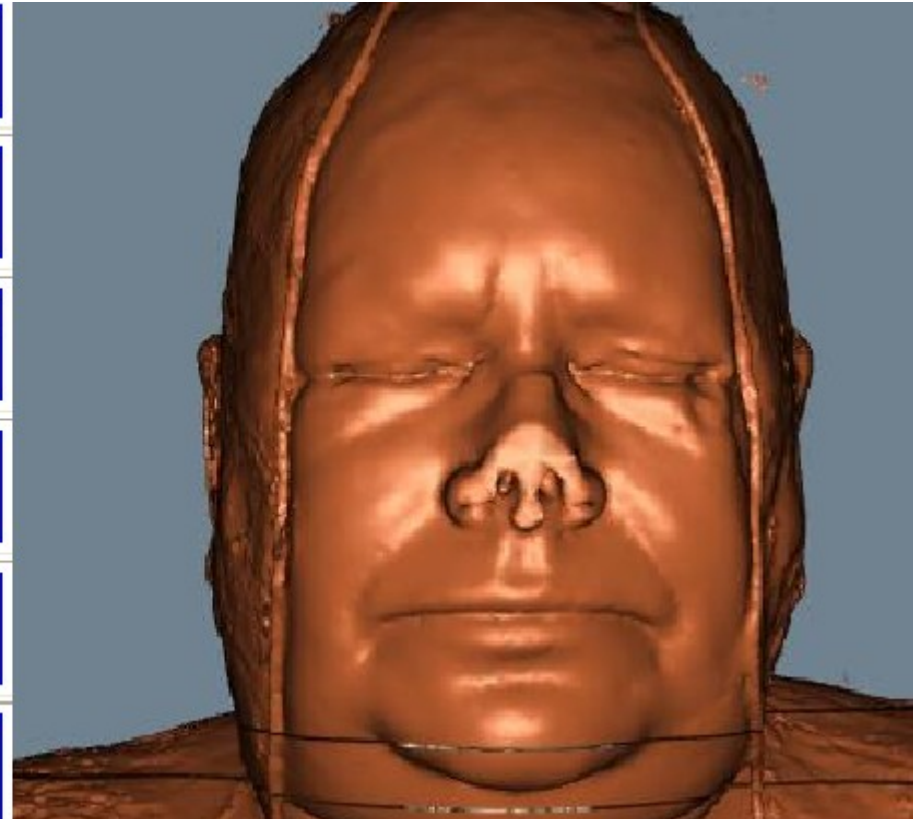
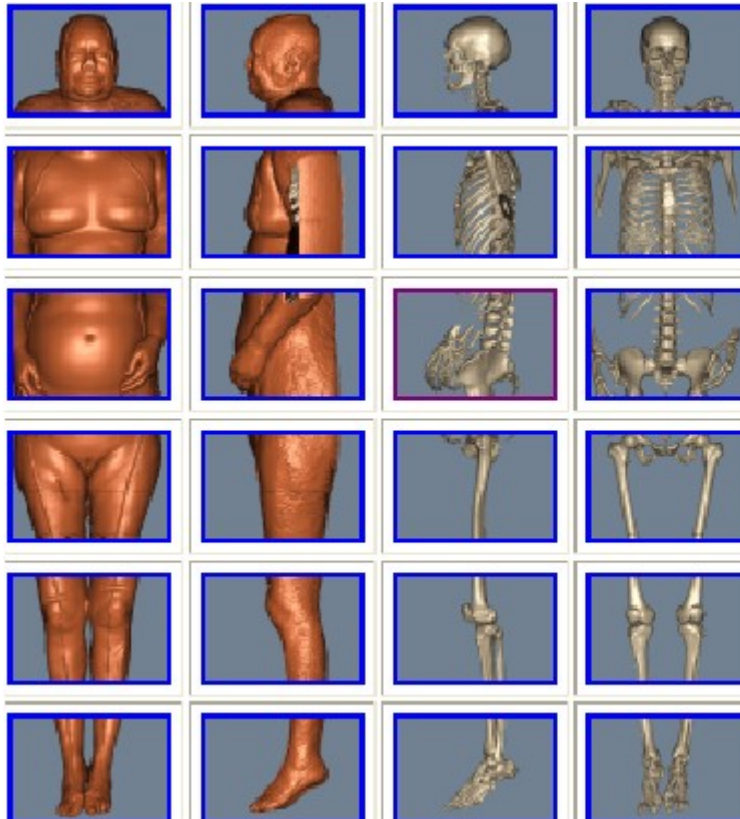
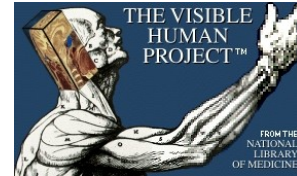
Indirect Volume Rendering

Goal: to extract a polygonal model from the voxel model and to render it.

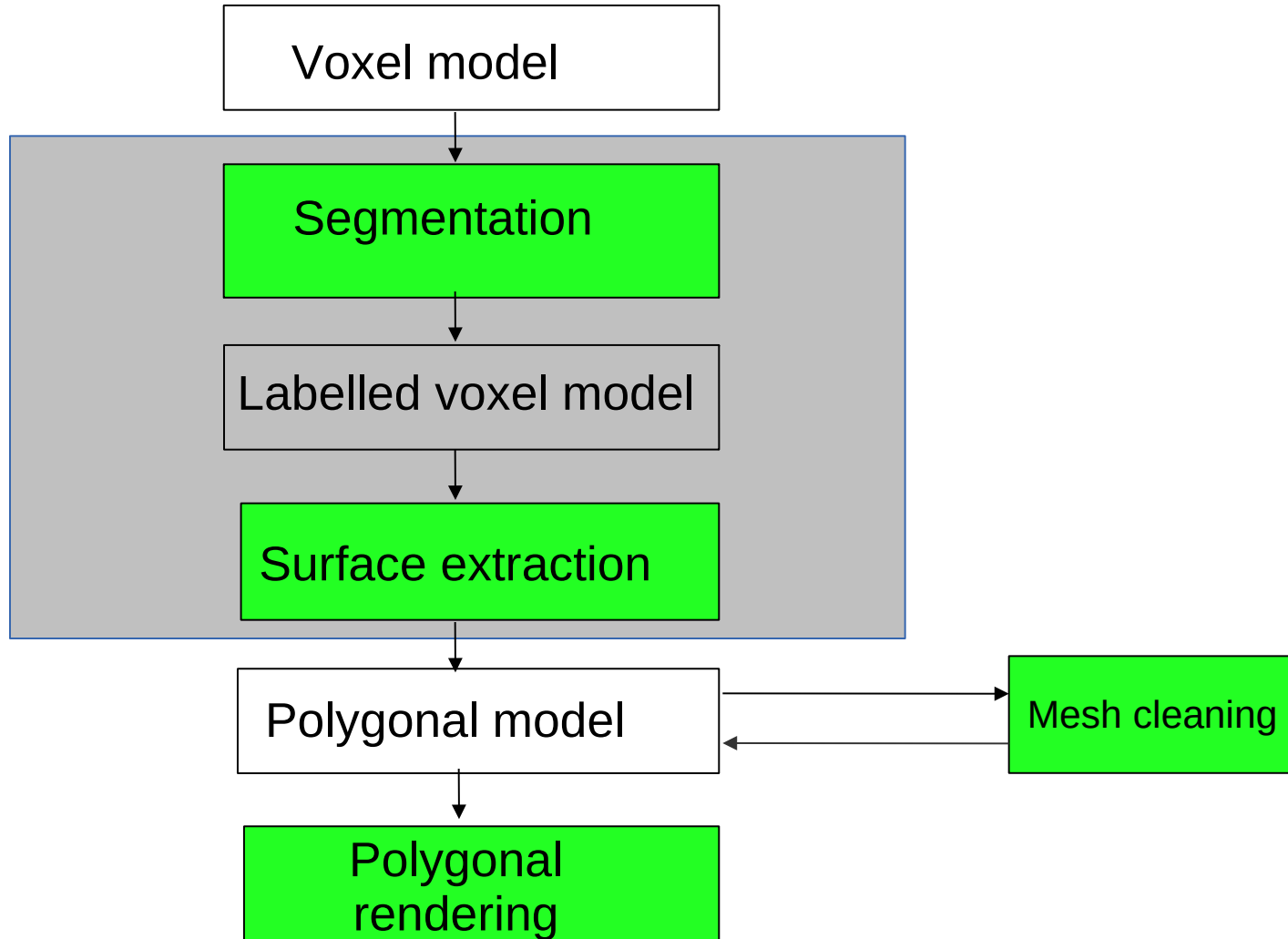


Indirect Volume Rendering

Skin and skeleton of the “visible woman”



Indirect volume rendering



Surface extraction

Methods

- 2D :
 - Contour extraction + tiling

- 3D :
 - “Block-form ” models: cuberille method
 - “Beveled-form” models: marching-cubes method
 - Point-based models: dividing-cubes method

Marching Cubes

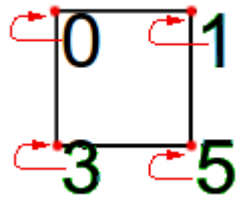
2D: Marching squares

0	1	3	4	4	3
3	4	9	7	6	4
4	6	8	6	4	3
1	2	6	4	3	2
0	2	4	3	3	2




Boundary surface: $V \geq 5$

Marching Cubes

2D: Marching squares



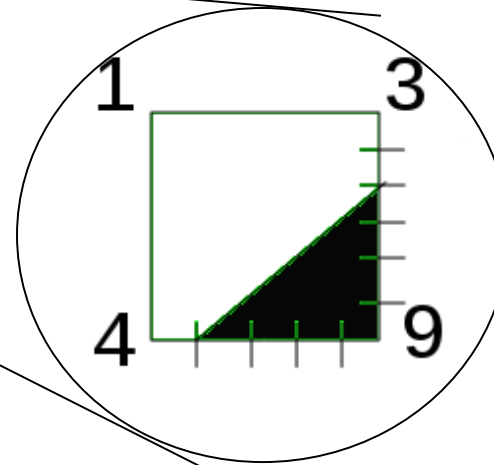
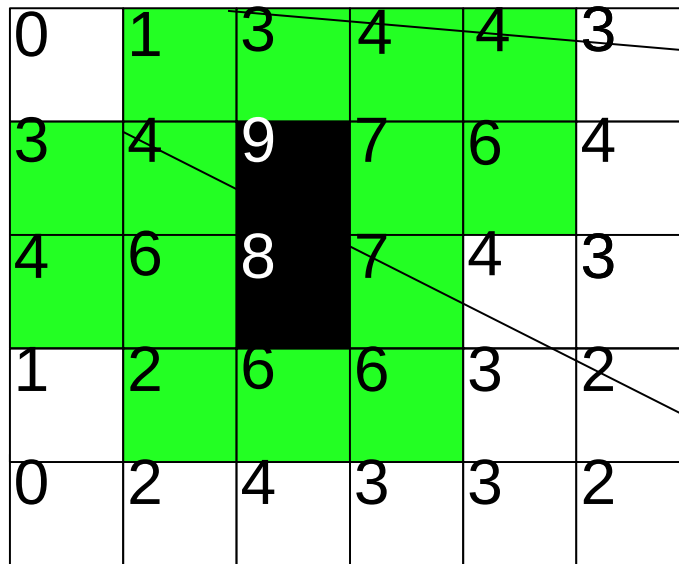
0	1	3	4	4	3
3	5	9	7	6	4
4	6	8	7	4	3
1	2	6	6	3	2
0	2	4	3	3	2




-  Inside
-  Outside
-  Boundary

Boundary surface: $V \geq 5$

Marching Cubes

2D: Marching squares

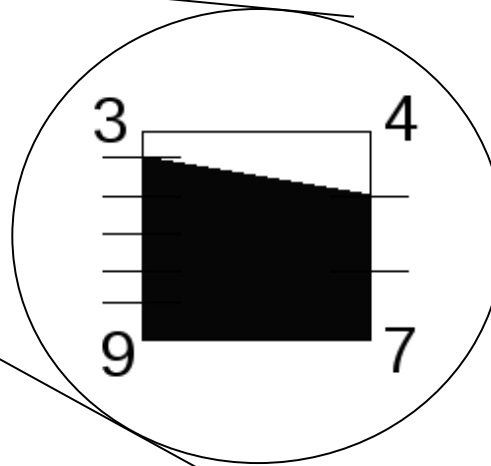
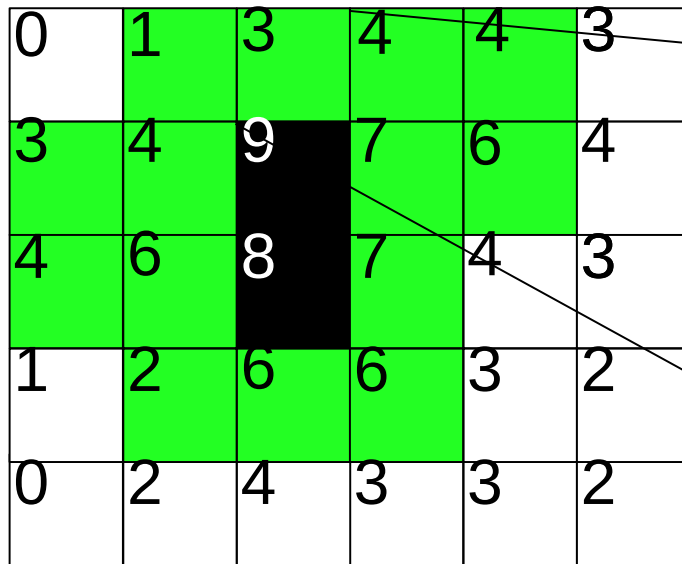





-  Inside
-  Outside
-  Boundary

Boundary surface: $V \geq 5$

Marching Cubes

2D: Marching squares

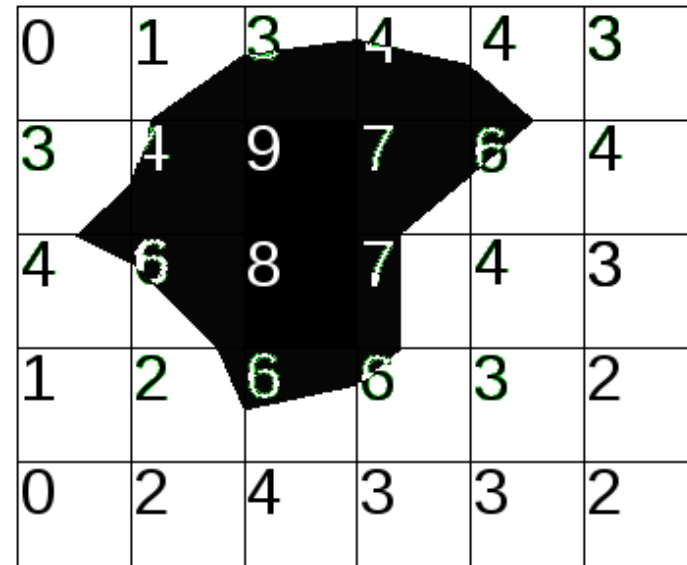
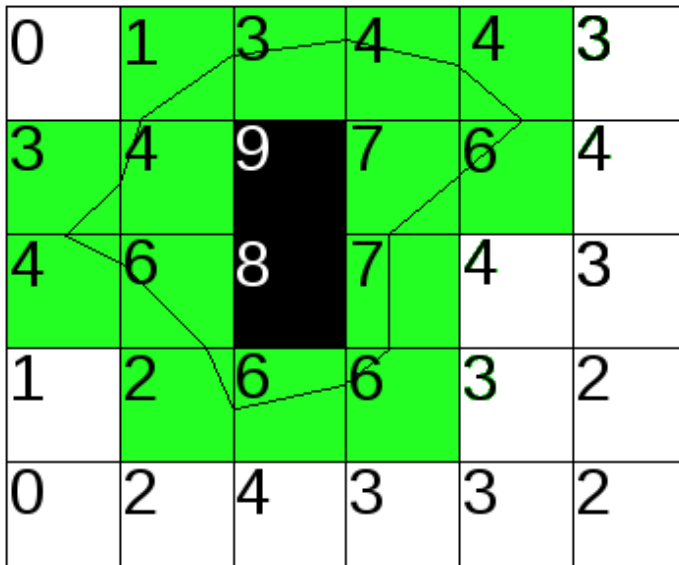


-  Inside
-  Outside
-  Boundary

Boundary surface: $V \geq 5$

Marching Cubes

2D: Marching squares

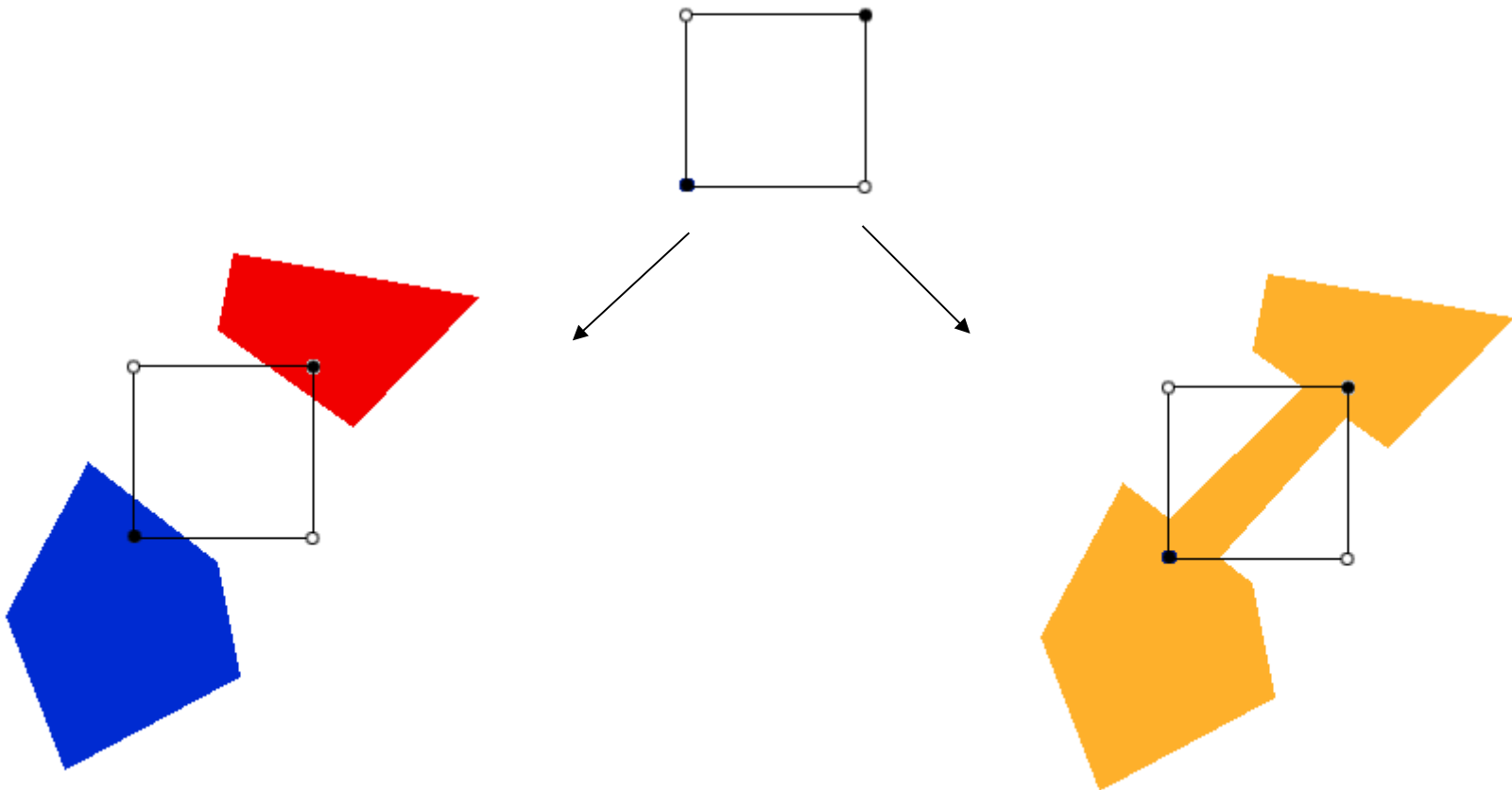


Boundary surface: $V \geq 5$

Marching Cubes

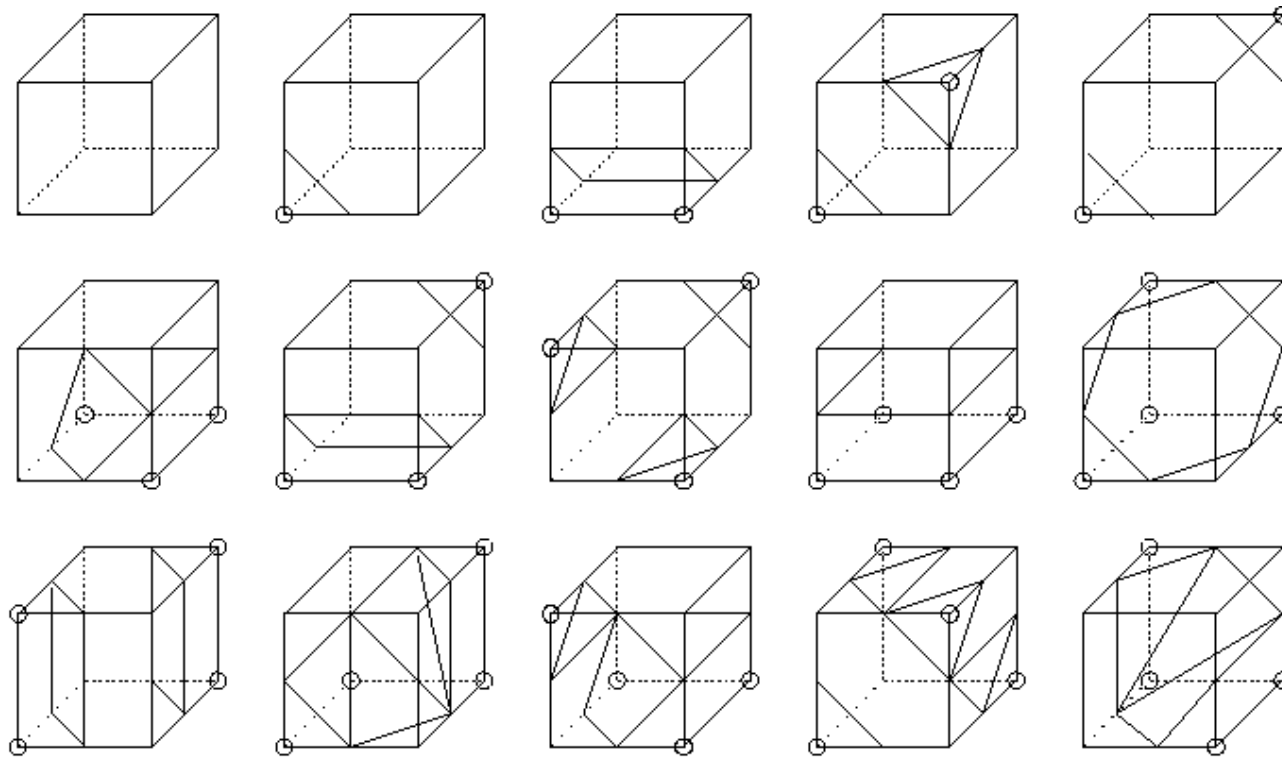
2D: marching squares

Ambiguous cases



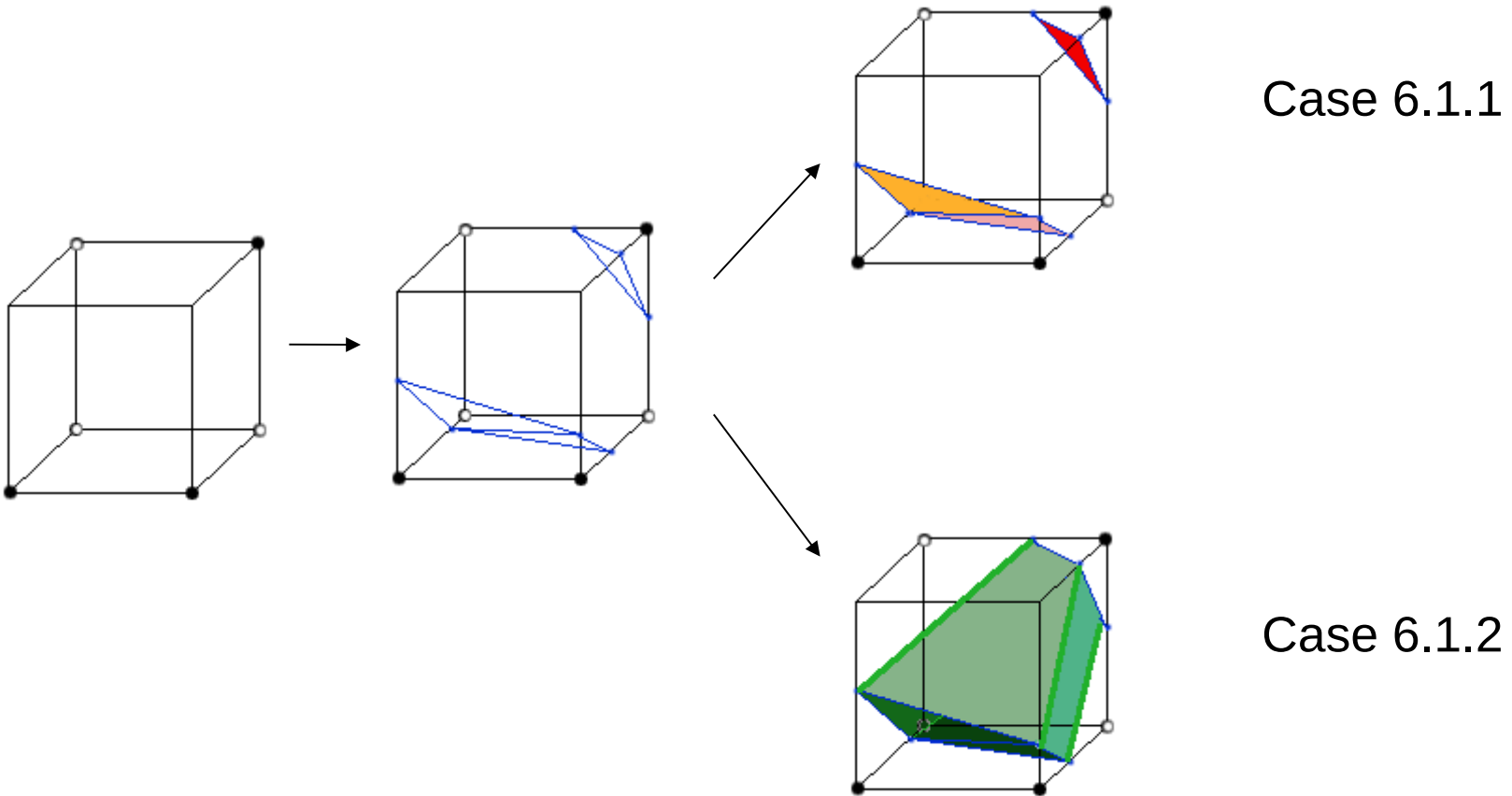
Marching Cubes (Lorensen i Cline 87)

Typification of the configuration of the surface inside a voxel (256 cases = 14 +1 case)



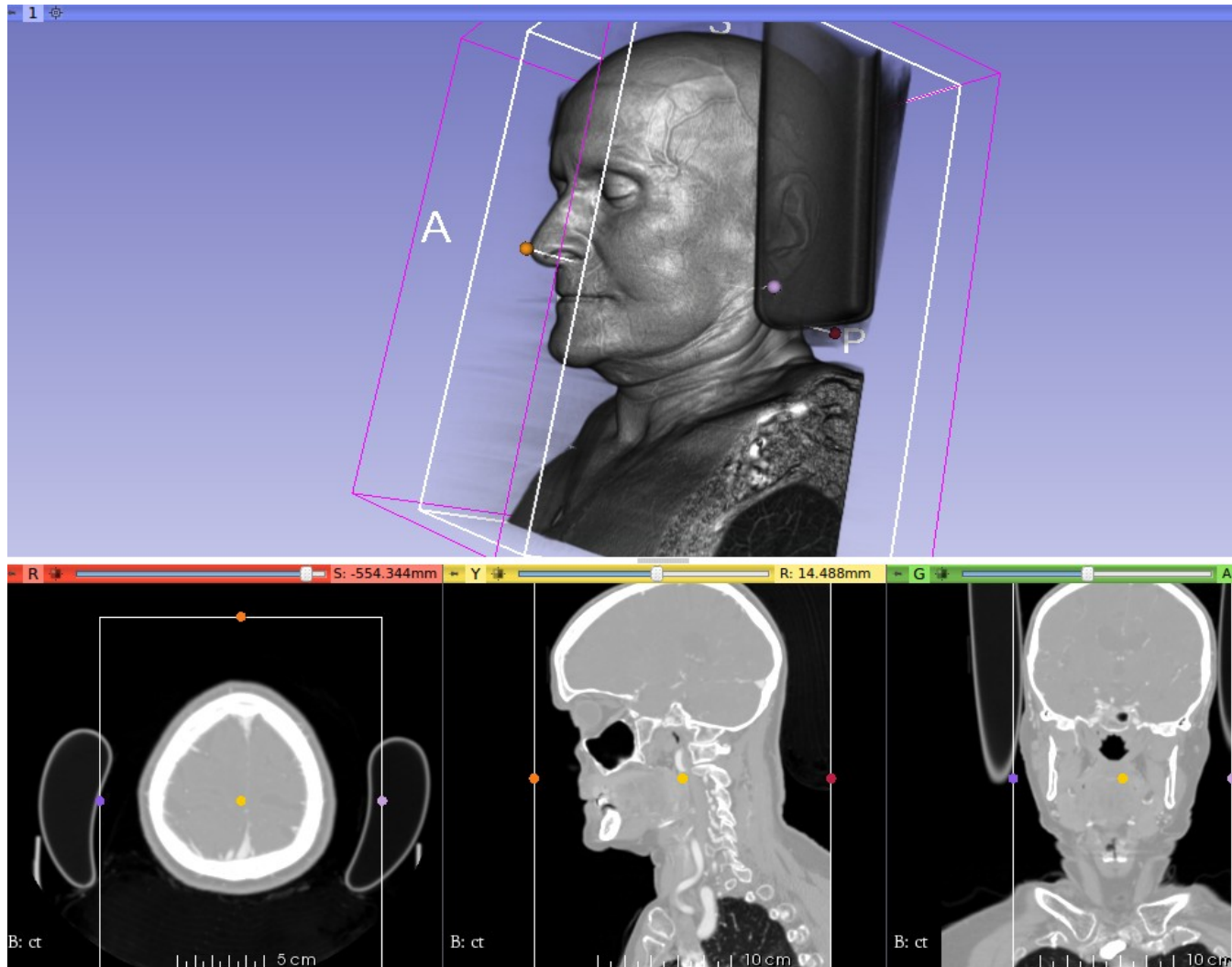
Marching Cubes (Lorensen and Cline 87)

Complementary cases

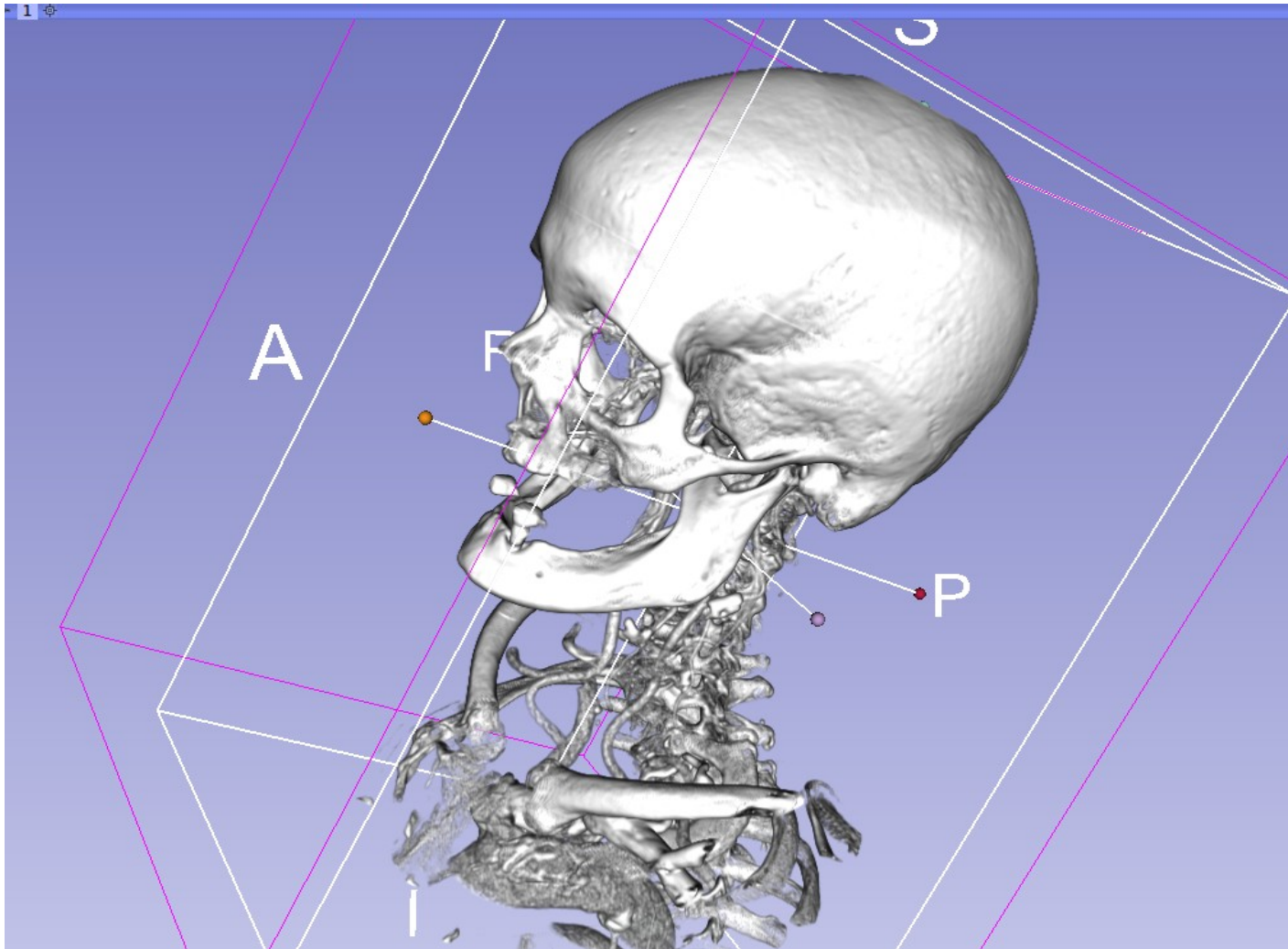


Practice

Explore CT data

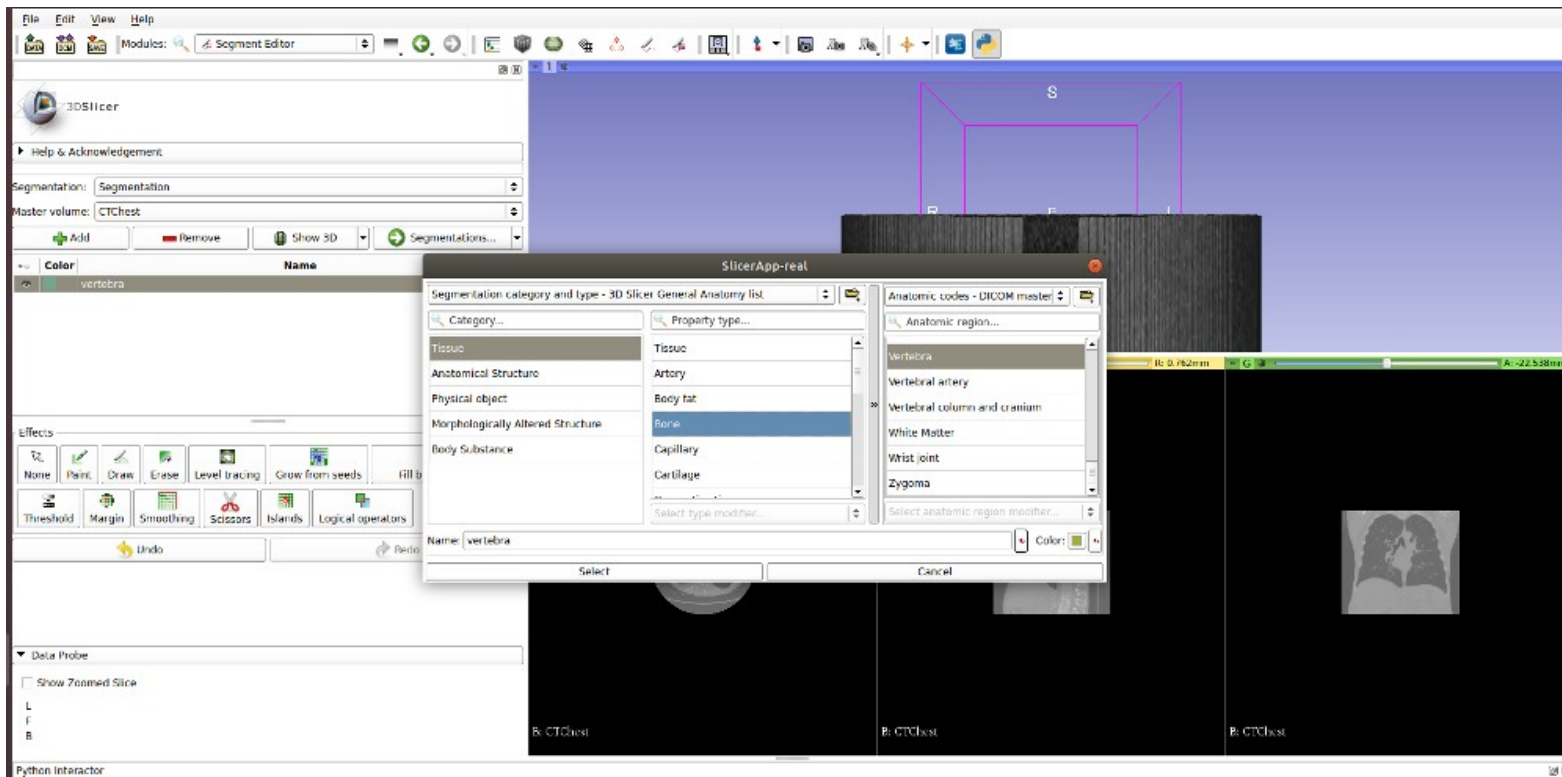


Explore the skull



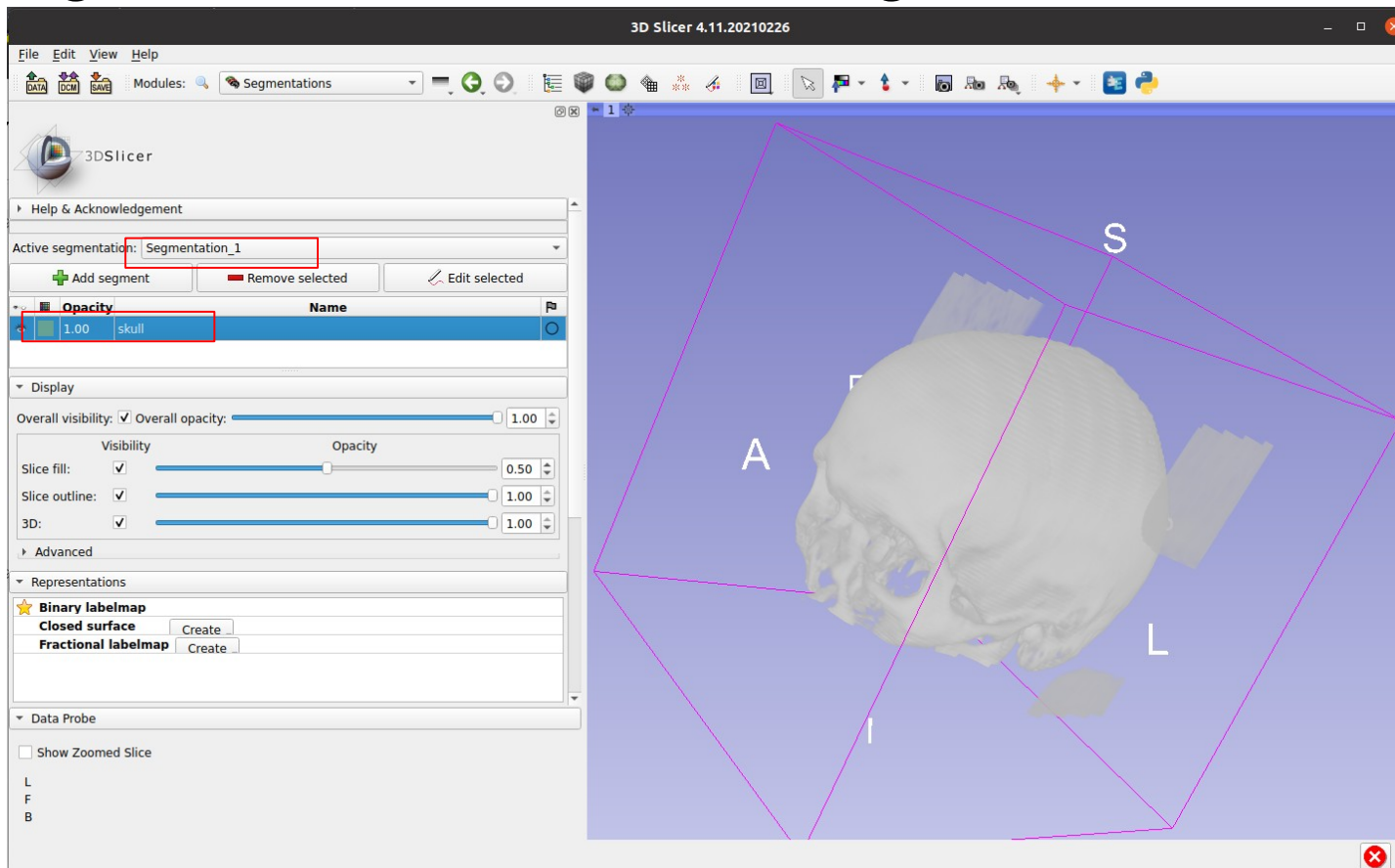
Use segmentation tools

In the menu segment editor create a new segmented region



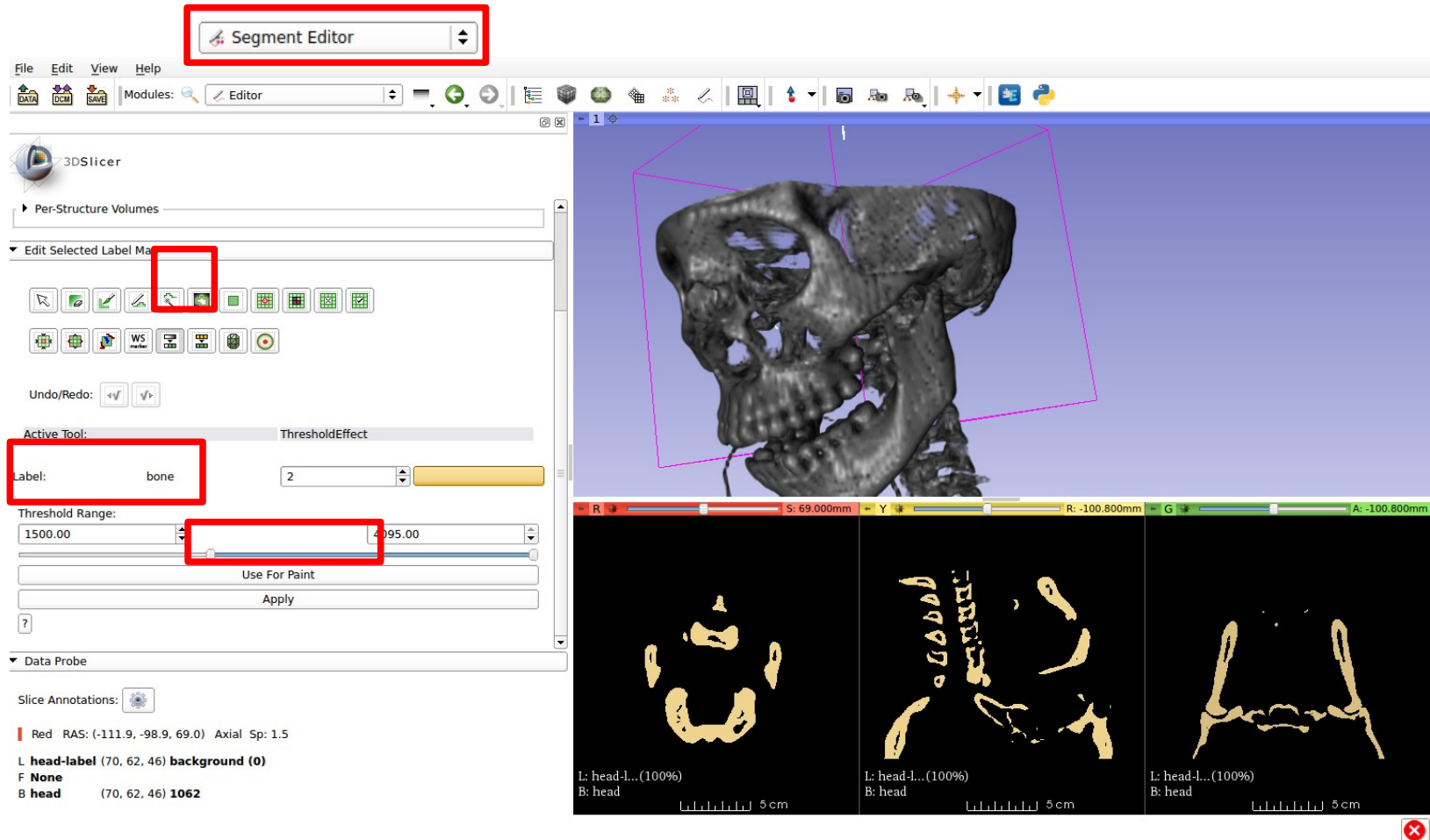
Use segmentation tools

Open the menu segmentations and create a new segmentation and a new segment called skull

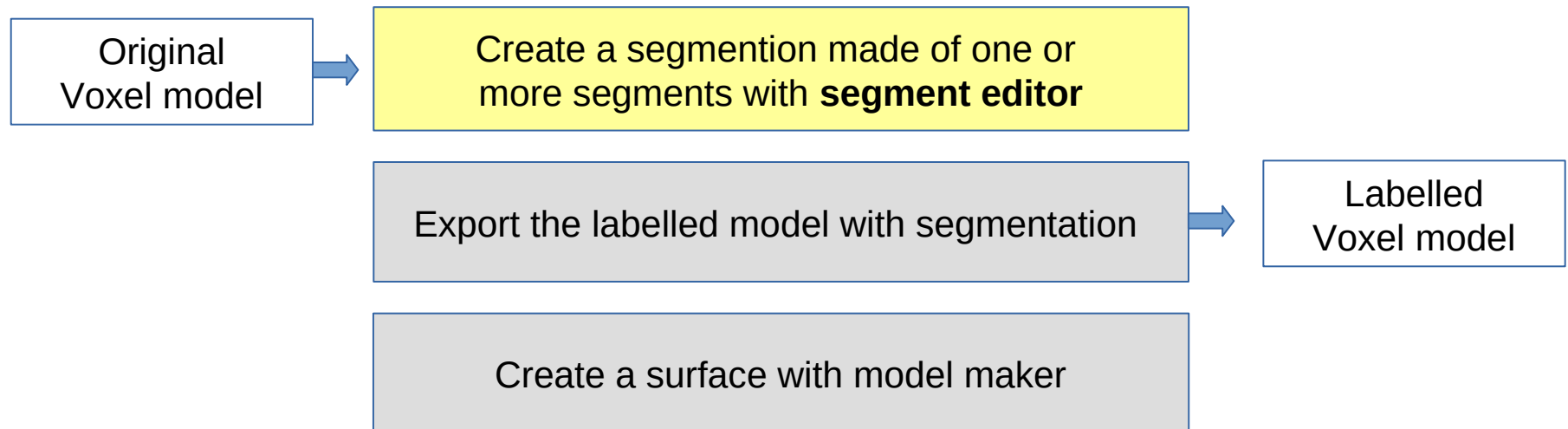


Extract the skull surface

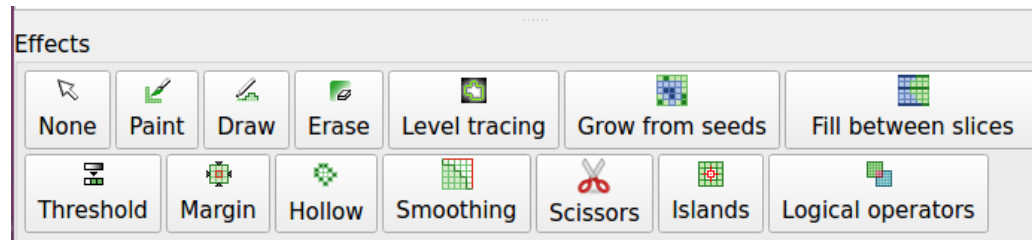
First segment by value. Edit the segment skull and in the segment editor panel choose the threshold



Indirect Volume rendering

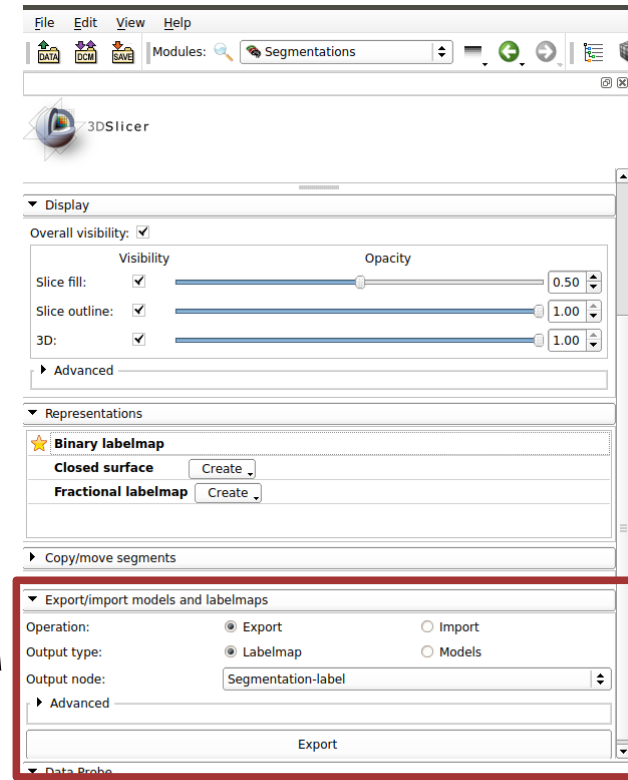
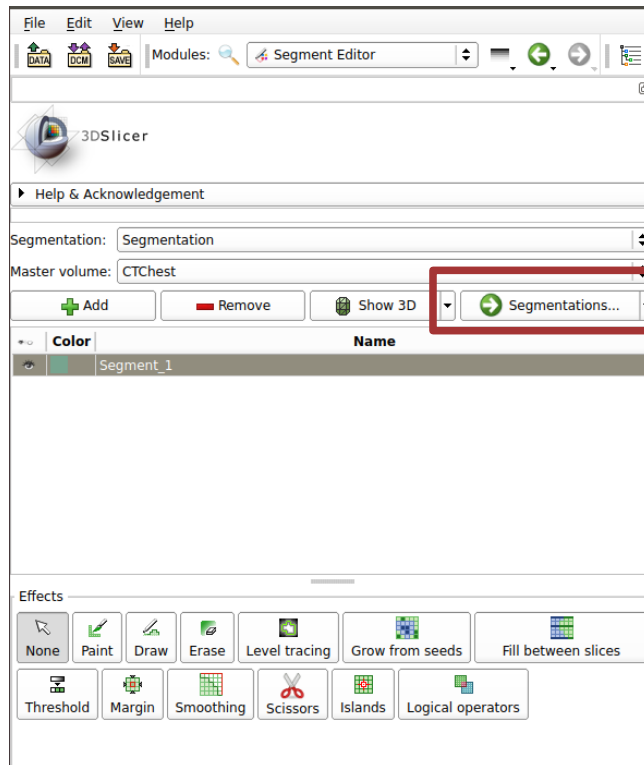


Tools



Extract the skull surface

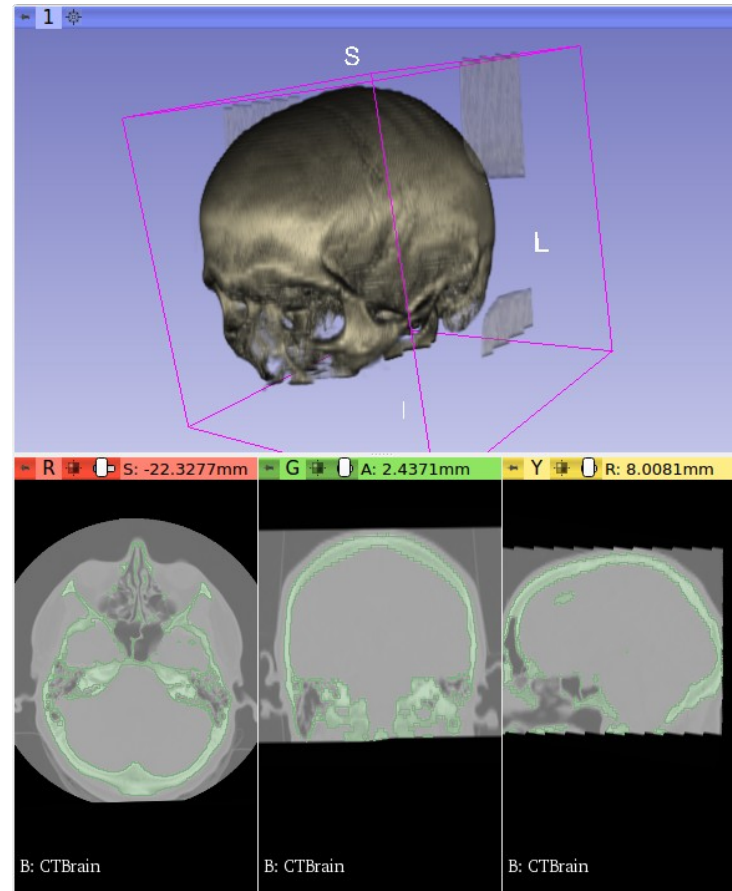
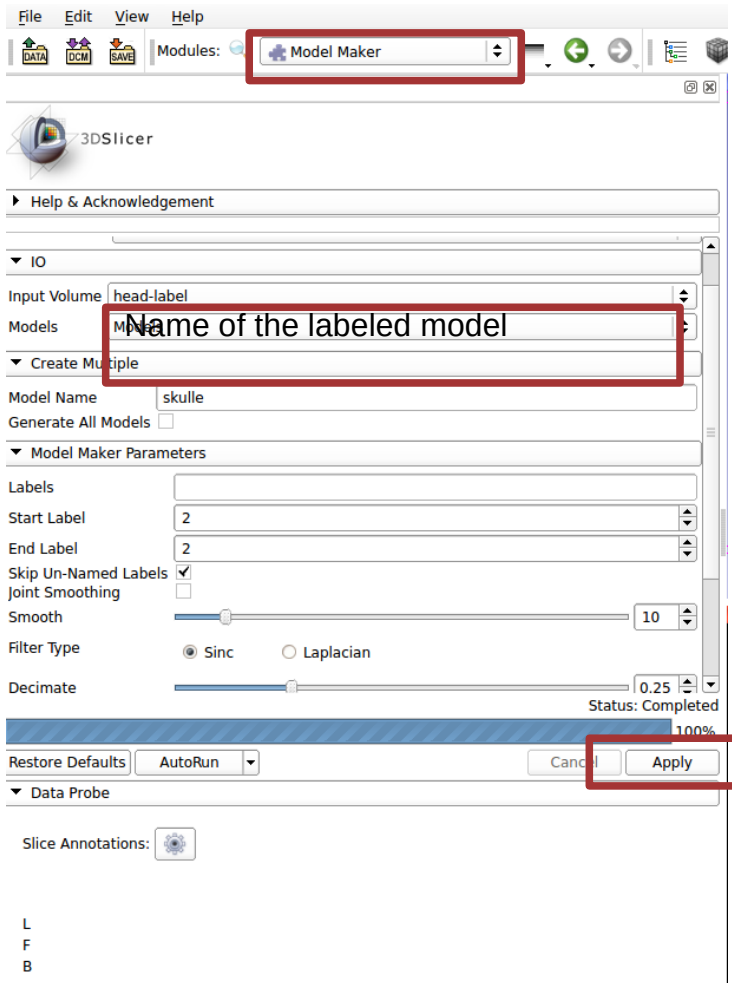
Next export the labeled volume



| Red (R 102.8, A 63.7, I 175.3) Axial Sp: 2.5
 L None
 F None
 B CTchest (121, 142, 69) -14

Extract the skull surface

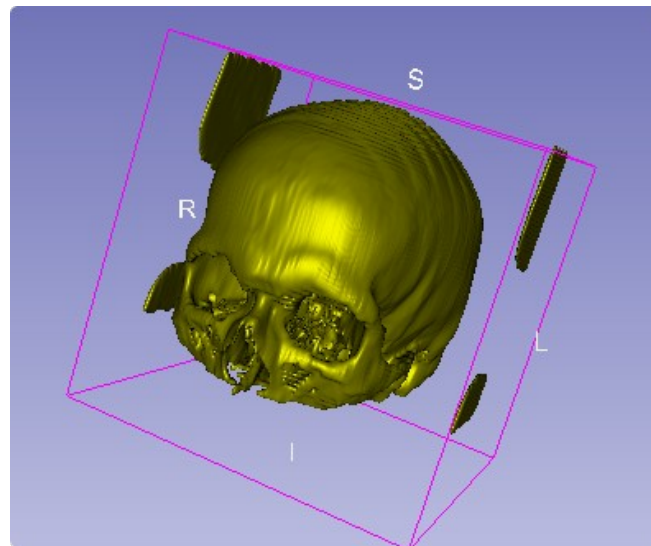
Next, create the surface model



Render the surface model

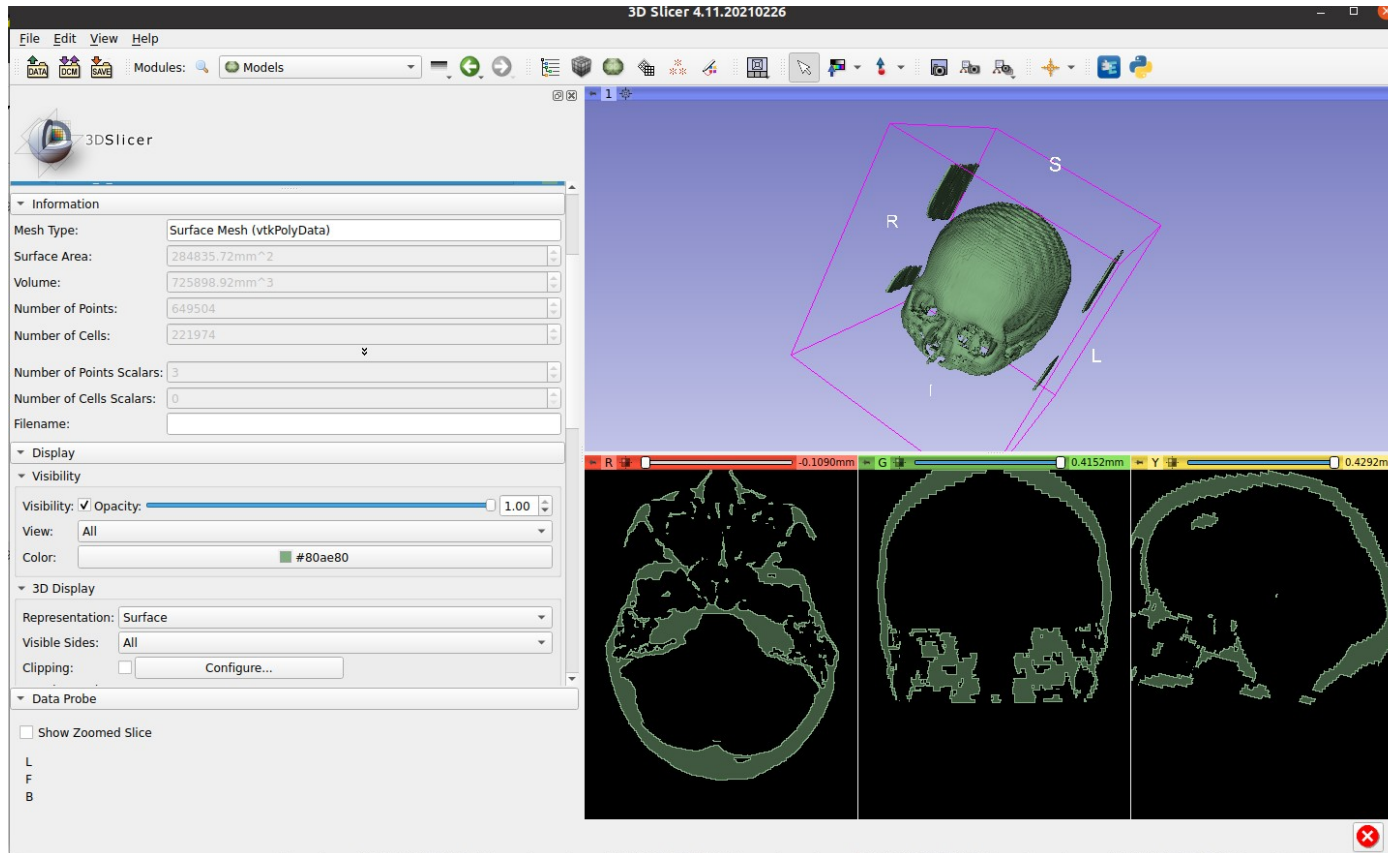
In Segmentations export to Labeled model. In Volume rendering turn of rendering of CTBrain and on rendering of the labelled voxel model.

In Segmentation export to Surface. Turn off volume rendering and in Models, play with shading parameters.



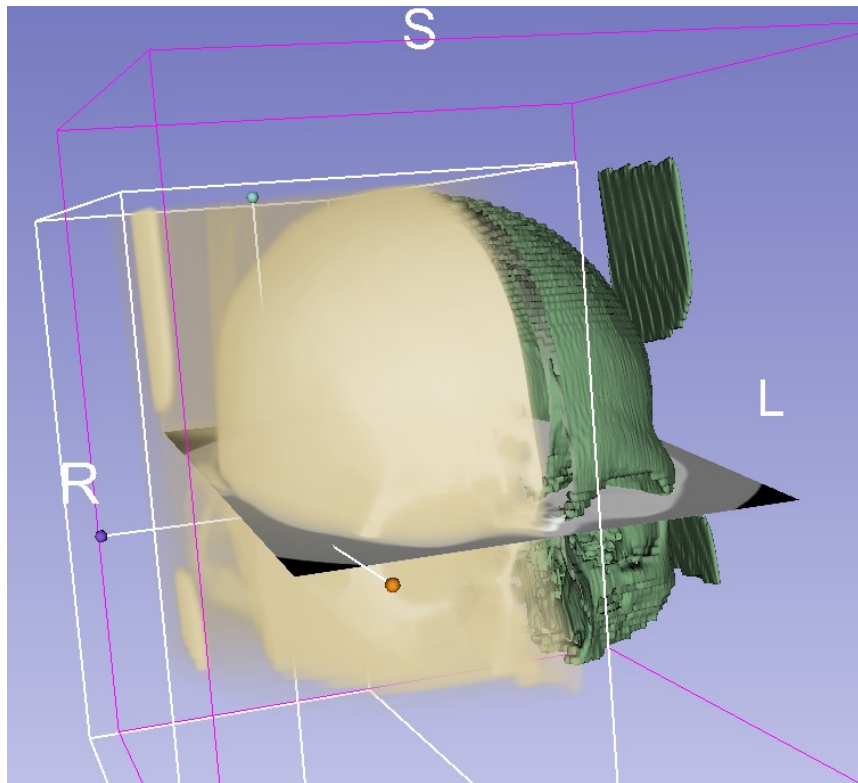
Analyze the surface model

In the menu model, select the created surface and analyze its characteristics

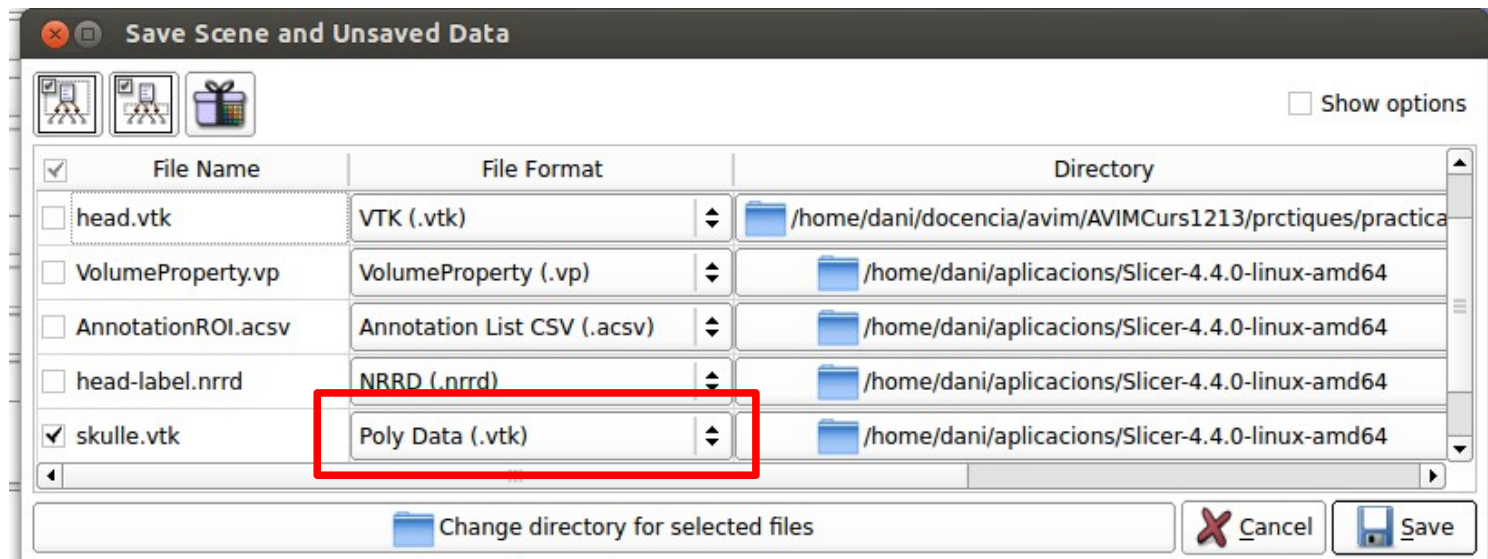


Render volume and surface

Find a suitable transfer function and render the volume and the surface. Use the clipping planes for the surface.



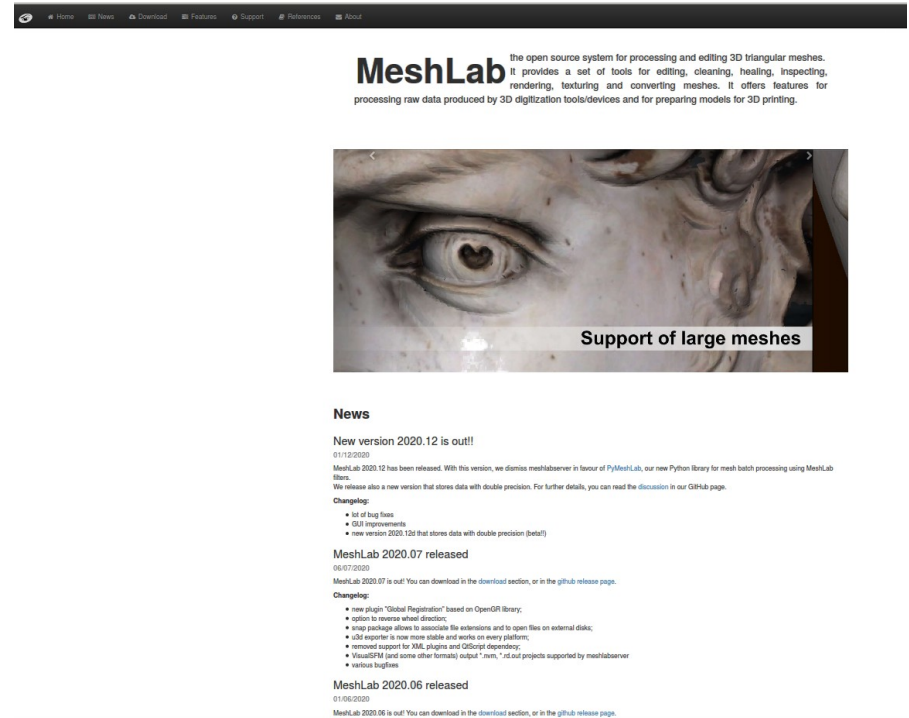
Save the surface model



Clean and simplify the surface

Surface Toolbox

▾ Reload & Test	
Reload	Reload and Test
Restart Slicer	
Edit	
Edit UI	
▾ Surface Toolbox	
Input model:	Segment_1
Output model:	Model_1
Decimation	
Smoothing	
Normals	
Mirror	
Cleaner	
Fill holes	
Connectivity	
Scale mesh	
Translate mesh	
Relax polygons	
Borders out	
Translate center to origin	
Toggle models	
Apply	



MeshLab the open source system for processing and editing 3D triangular meshes. It provides a set of tools for editing, cleaning, healing, inspecting, rendering, texturing and converting meshes. It offers features for processing raw data produced by 3D digitization tools/devices and for preparing models for 3D printing.

Support of large meshes

News

New version 2020.12 is out!!
01/12/2020
MeshLab 2020.12 has been released. With this version, we deprecate meshlabserver in favour of PyMeshLab, our new Python library for mesh batch processing using MeshLab filters. We release also a new version that stores data with double precision. For further details, you can read the [discussion](#) in our GitHub page.

Changelog:

- lot of bug fixes
- GUI improvements
- new version 2020.12a that stores data with double precision (beta!)

MeshLab 2020.07 released
06/07/2020
MeshLab 2020.07 is out! You can download in the [download section](#), or in the [github release page](#).

Changelog:

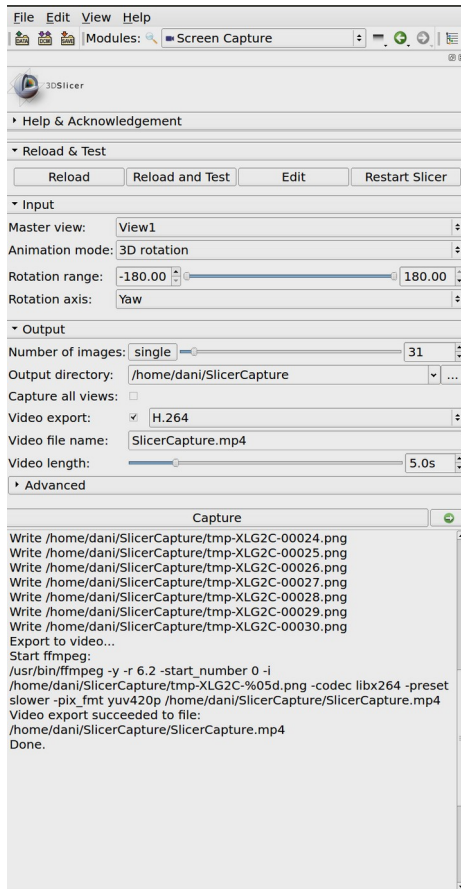
- new plugin "Global Registration" based on OpenQR library;
- option to reverse wheel direction;
- snap package allows to associate file extensions and to open files on external disks;
- GUI exports to new more stable and works on every platform;
- removed support for XAL plugins and Qt5.9.5 dependency;
- VisualSM (and some other formats) output "norms" not out projects supported by meshlabserver
- various bugfixes

MeshLab 2020.06 released
01/06/2020
MeshLab 2020.06 is out! You can download in the [download section](#), or in the [github release page](#).

Create animations views

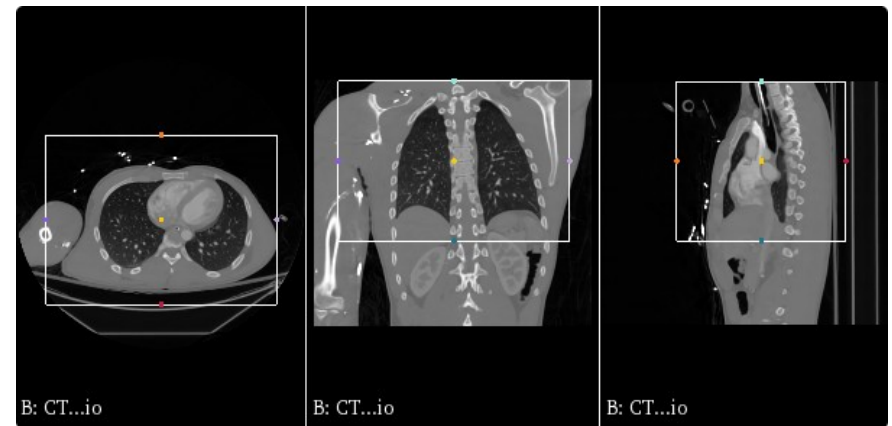
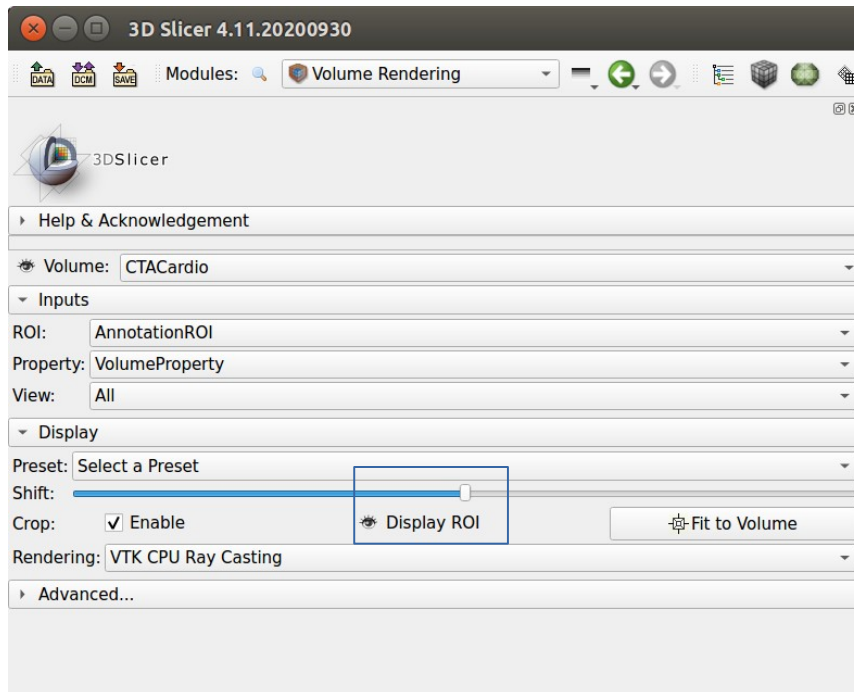
You may need to install ffmpeg. Check in :

https://slicer.readthedocs.io/en/latest/user_guide/modules/screenshot.html#setting-up-ffmpeg



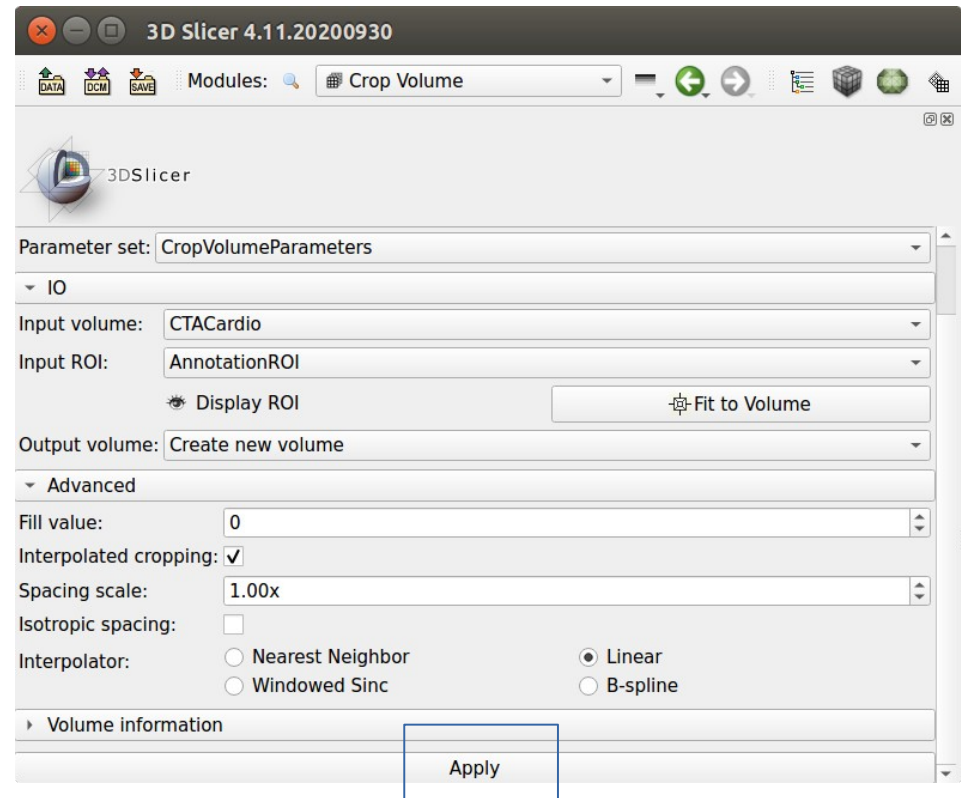
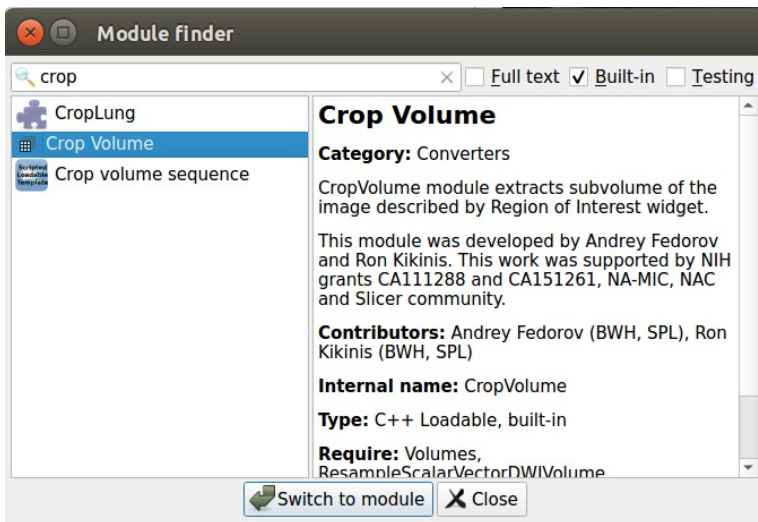
Crop

When dataset are huge crop them and work with a subset. In volume rendering press on Display Roi (Region of Interest). Move the planes (by pressing and moving the small balls) in 2D (faster) to define a smaller region.



Crop

Select the module Crop Volume. Check the parameters. Apply.

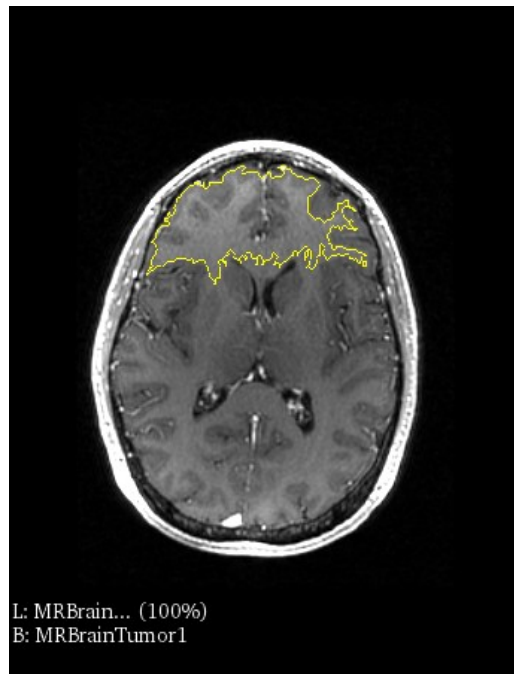


Explore more segmentation tools

Manual segmentation

Define contours on a slice.

Use LevelTracingEffect



▼ Edit Selected Label Map



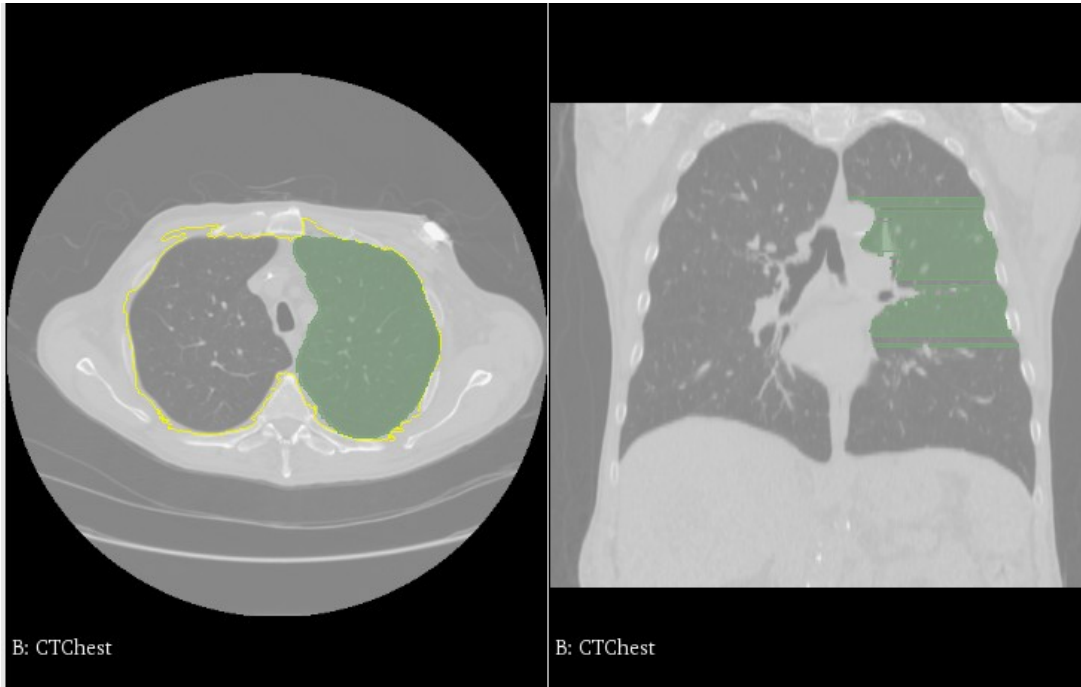
Undo/Redo: [undo] [redo]

Active Tool: LevelTracingEffect

Label: tissue 1 [color bar]

- Paint Over
- Threshold Paint
-

Manual segmentation



Define contours on a slice.

Use `LevelTracingEffect`

Move fast from slice to the other using:

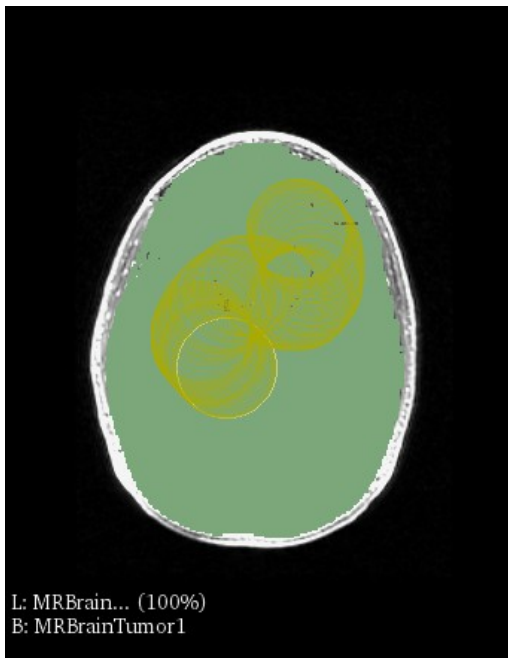
- f -> move to next slice
- b -> move to previous slice

Manual segmentation


Define contours on a slice.


Refine the segmentation manually with Draw and Paint Effects.

Select the Erase Color to erase while painting




▼ Edit Selected Label Map



Undo/Redo: 

Active Tool: LevelTracingEffect

Label: tissue 1


Paint Over
 Threshold Paint



Manual segmentation


You can change the name and value of a label any time with edit selected map. This is useful to merge structures



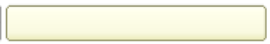

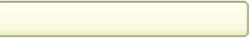
▼ Edit Selected Label Map






Undo/Redo: 

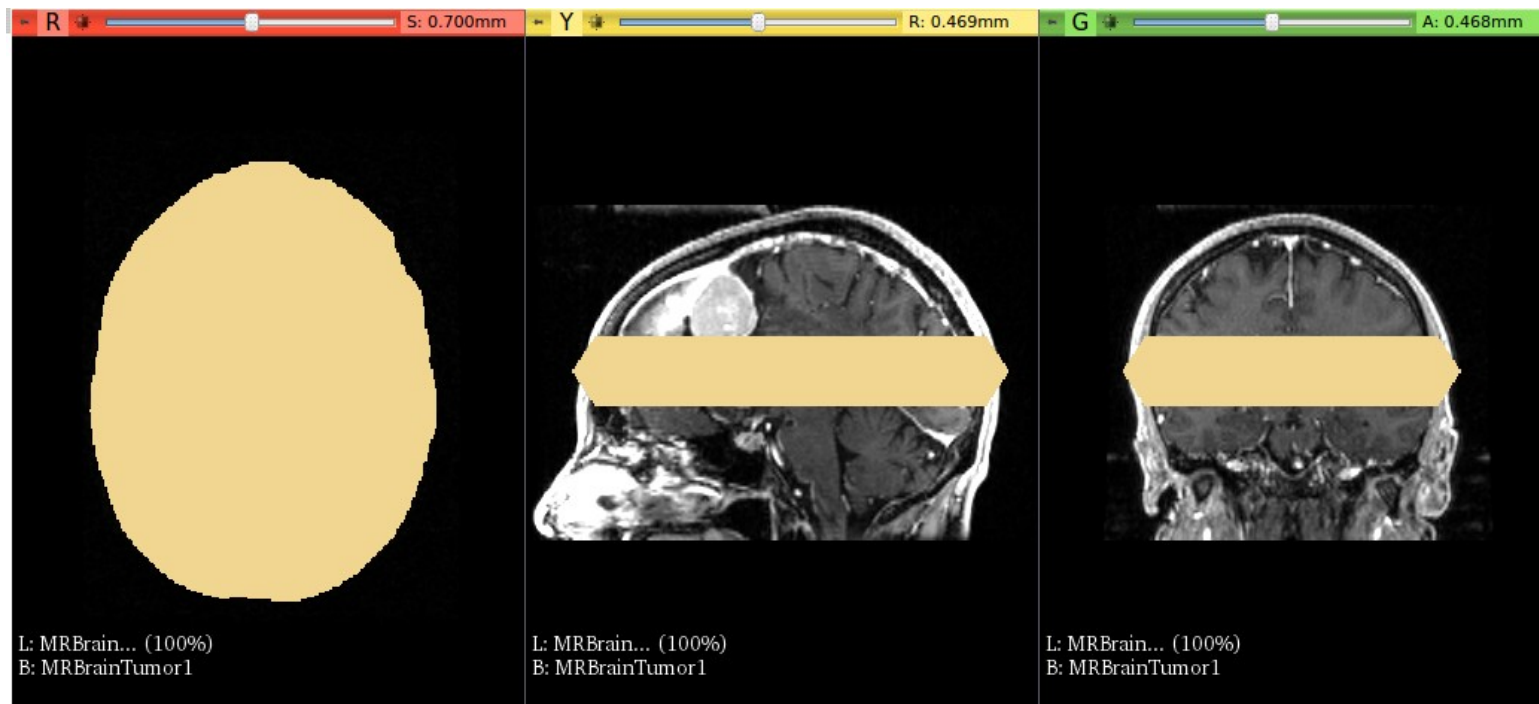
Active Tool: ChangeLabelEffect

Label:	brain	38	
Input Color:	tissue	1	
Output Color:	brain	38	



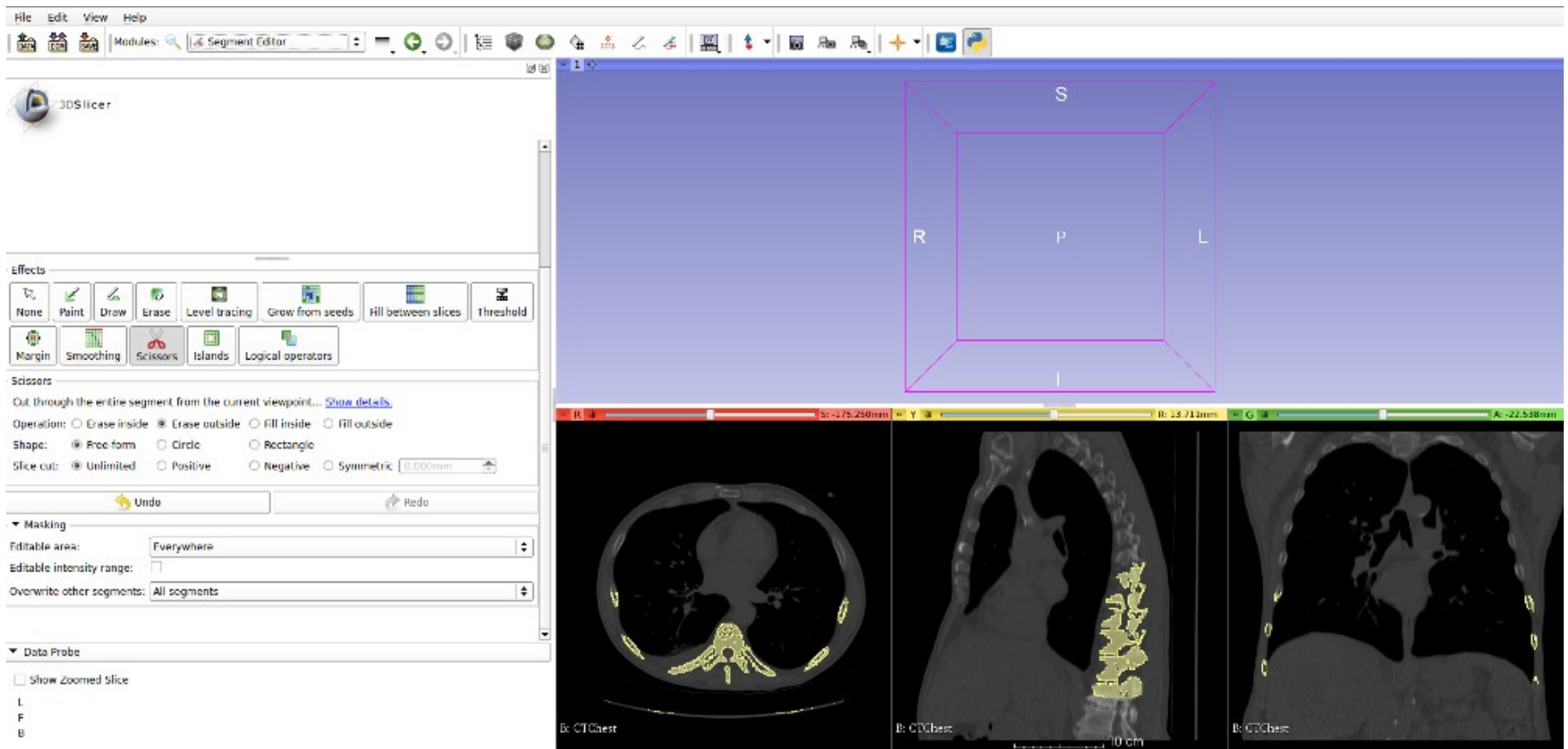
Manual segmentation

You can use Erode/Dilate to expand/reduce a segmentation intra slice and between slices



Manual segmentation

In the menu segment editor use the scissor tool to segment only what you want. Then remove the islands.



Semi-automatic segmentation

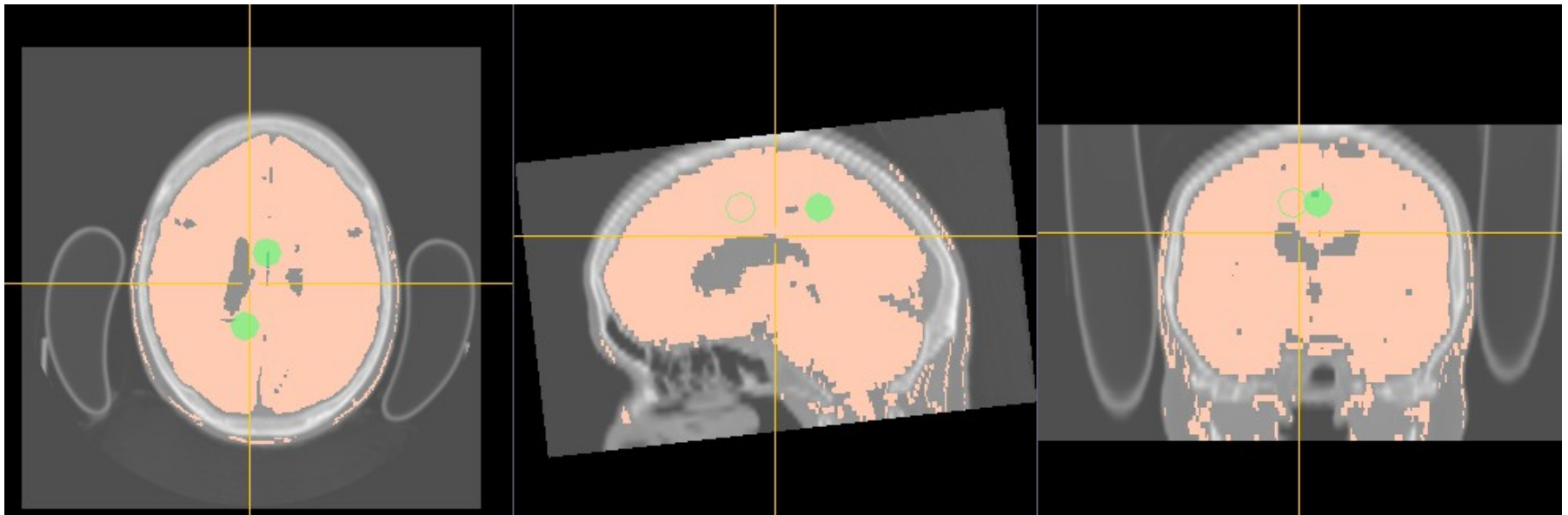
Method:

- Put some fiducials in the region of interest
- Try Simple Region Growing segmentation
- Refine with manual segmentation

Markups

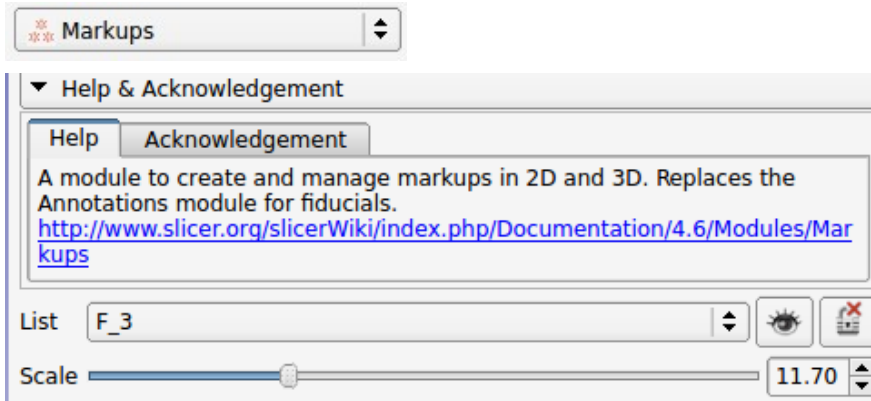
Simple Region Growing Segmentation

Editor

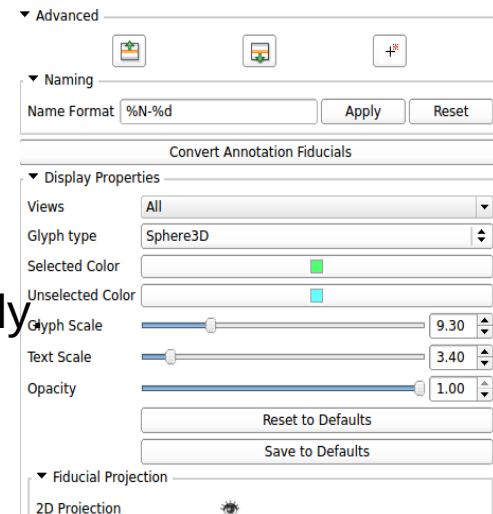


Add Fiducials

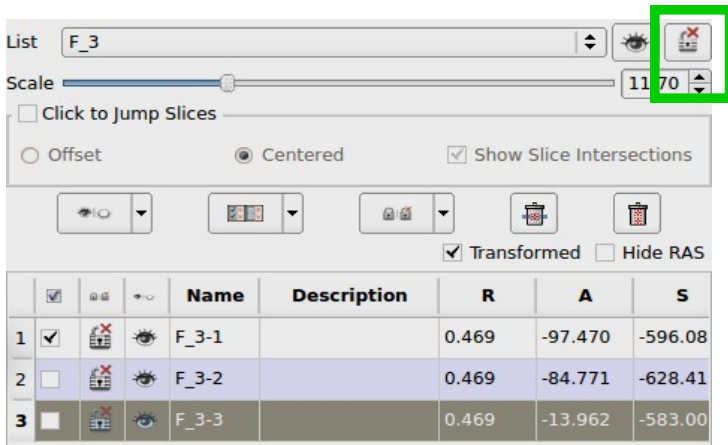
Create a new fiducial list



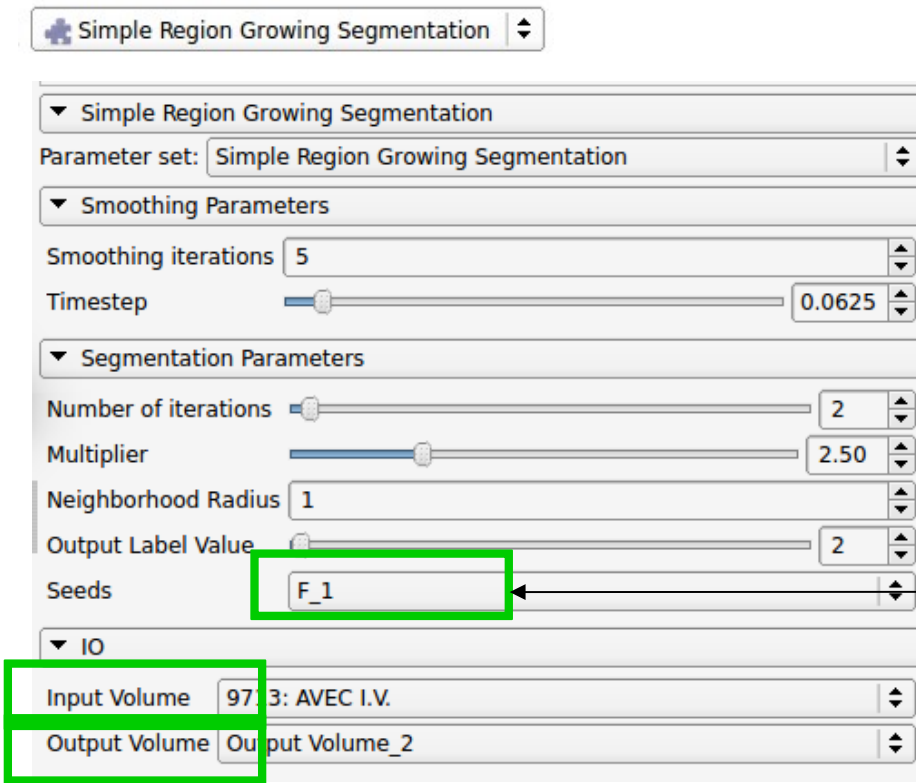
Define the visual aspect of the fiducials (or markers): color, glyph, size....



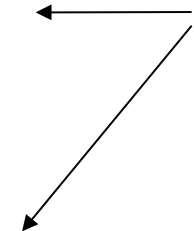
Add fiducials and modify them. Move them interactively



Simple region growing segmentation



Region growing parameters

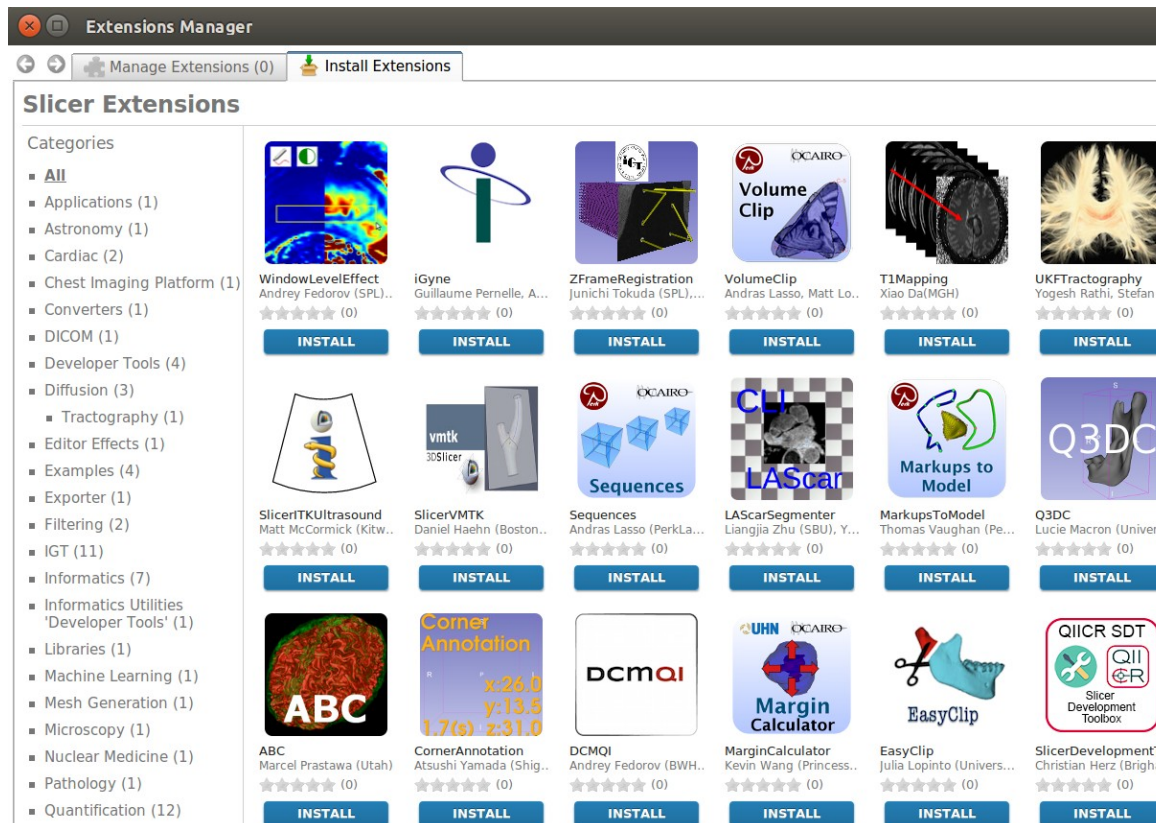


List of fiducials used.



Look for a suitable extension

In the extension manager, you can find many extensions that may be useful for you



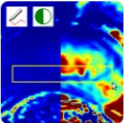

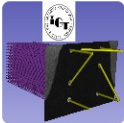
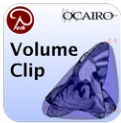








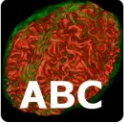





Extensions Manager

Manage Extensions (0) Install Extensions

Slicer Extensions

Categories

- All
- Applications (1)
- Astronomy (1)
- Cardiac (2)
- Chest Imaging Platform (1)
- Converters (1)
- DICOM (1)
- Developer Tools (4)
- Diffusion (3)
 - Tractography (1)
- Editor Effects (1)
- Examples (4)
- Exporter (1)
- Filtering (2)
- IGT (11)
- Informatics (7)
- Informatics Utilities 'Developer Tools' (1)
- Libraries (1)
- Machine Learning (1)
- Mesh Generation (1)
- Microscopy (1)
- Nuclear Medicine (1)
- Pathology (1)
- Quantification (12)

 WindowLevelEffect Andrey Fedorov (SPL)...	 iGyne Guillaume Pernelle, A...	 ZFrameRegistration Junichi Tokuda (SPL)...	 VolumeClip Andras Lasso, Matt Lo...	 T1Mapping Xiao Da(MGH)	 UKFTractography Yogesh Rathi, Stefan ...
★★★★★ (0)	★★★★★ (0)	★★★★★ (0)	★★★★★ (0)	★★★★★ (0)	★★★★★ (0)
INSTALL	INSTALL	INSTALL	INSTALL	INSTALL	INSTALL
 SlicerITKUltrasound Matt McCormick (Kitw...	 SlicerVMTK Daniel Haehn (Boston...	 Sequences Andras Lasso (PerkLa...	 LAScarSegmenter Liangjia Zhu (SBU), Y...	 MarkupsToModel Thomas Vaughan (Pe...	 Q3DC Lucie Macron (Univer...
★★★★★ (0)	★★★★★ (0)	★★★★★ (0)	★★★★★ (0)	★★★★★ (0)	★★★★★ (0)
INSTALL	INSTALL	INSTALL	INSTALL	INSTALL	INSTALL
 ABC Marcel Prastawa (Utah)	 CornerAnnotation Atsushi Yamada (Shig...	 DCMQI Andrey Fedorov (BWH...	 MarginCalculator Kevin Wang (Princess...	 EasyClip Julia Lopinto (Univers...	 SlicerDevelopmentT... Christian Herz (Brigha...
★★★★★ (0)	★★★★★ (0)	★★★★★ (0)	★★★★★ (0)	★★★★★ (0)	★★★★★ (0)
INSTALL	INSTALL	INSTALL	INSTALL	INSTALL	INSTALL

Example

The airway segmentation extension by Pietro Nardelli.

Install it with the extension manager. You will need to restart Slicer

Follow the tutorial at:



Test it with CT Chest (see next slide)

<https://www.slicer.org/wiki/Documentation/Nightly/Modules/AirwaySegmentation>

Example

Select the new module. Select the CT chest as input module and create a new segmented volume. Create a new seed putting a fiducial inside the trachea. Select linear reconstruction kernel and apply. Set the transfer function to make the segmented region visible.

