

This notebook is intended to demonstrate how select registration, segmentation, and image mathematical methods of ITKTubeTK can be combined to perform multi-channel brain extraction (aka. skull stripping for patient data containing multiple MRI sequences).

There are many other (probably more effective) brain extraction methods available as open-source software such as BET and BET2 in the FSL package (albeit such methods are only for single channel data). If you need to perform brain extraction for a large collection of scans that do not contain major pathologies, please use one of those packages. This notebook is meant to show off the capabilities of specific ITKTubeTK methods, not to demonstrate how to "solve" brain extraction.

```
In [1]: import itk
from itk import TubeTK as ttk

from itkwidgets import view

import numpy as np
import time
```

```
In [2]: ImageType = itk.Image[itk.F, 3]

ReaderType = itk.ImageFileReader[ImageType]

InputBaseName = "../Data/CTA-Head/CTA"

filename = InputBaseName + ".mha"
reader1 = ReaderType.New(FileName=filename)
reader1.Update()
im1 = reader1.GetOutput()
```

```
In [3]: im1iso = im1

# Our example data is already in isotropic voxels (has the same voxel spacing in
# So, we simply set im1iso = im. Otherwise, if your data is not isotropic,
# you should include the following commands:

#resamp = ttk.ResampleImage[ImageType].New(Input = im1)
#resamp.SetMakeHighResIso(True)
#resamp.Update()
#im1iso = resamp.GetOutput()

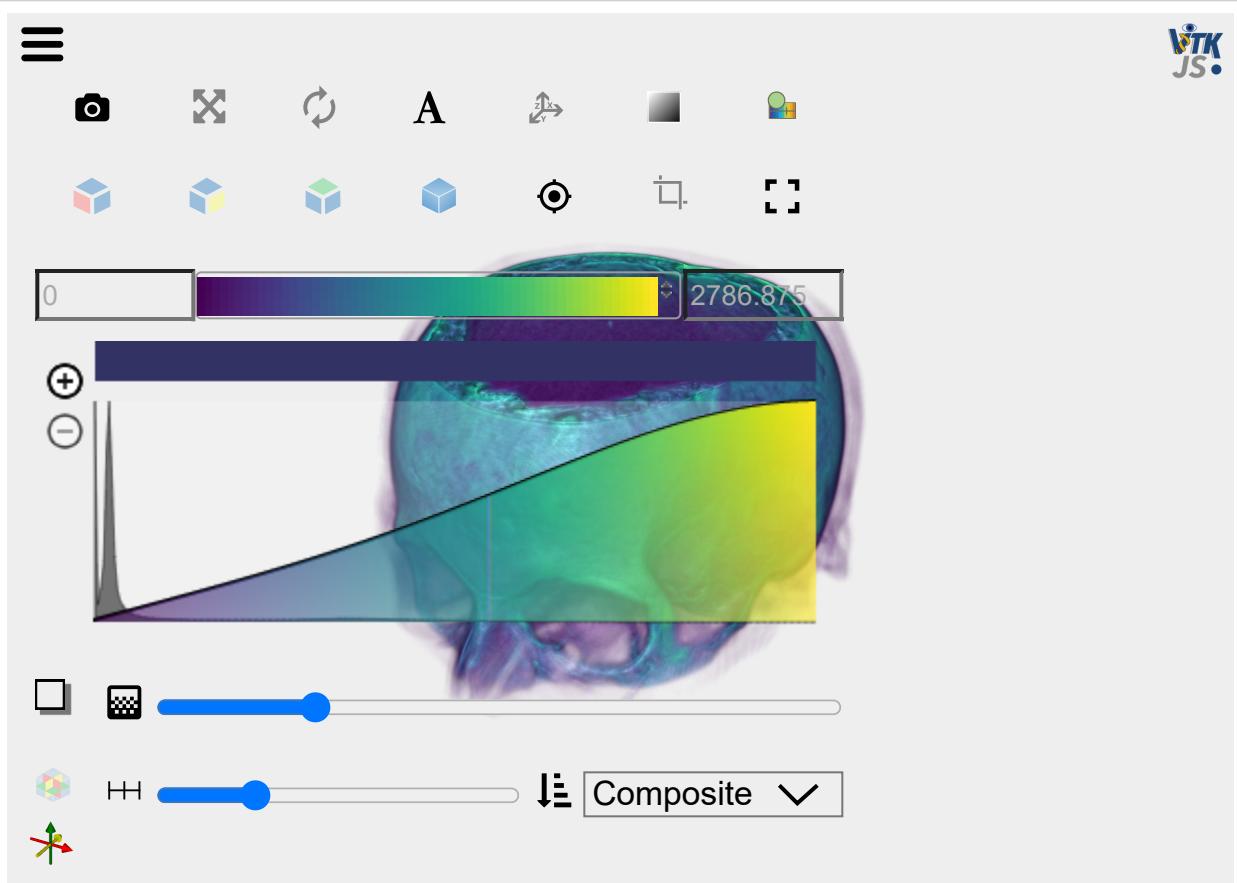
#filename = InputBaseName + "-Iso.mha"
#itk.imwrite(im1iso, filename)
```

Here we are performing multi-atlas segmentation. We will read 8 MRI image for which we already segmented the brain (brain segmentation is much easier for MRI FLASH images).

Those 8 MRI will be registered with the new patient, and their brain masks will be used to vote for an initial segmentation of the brain from the new patient. That initial segmentation will then be refined using a multi-channel connected components method - but for CTA we only have one channel.

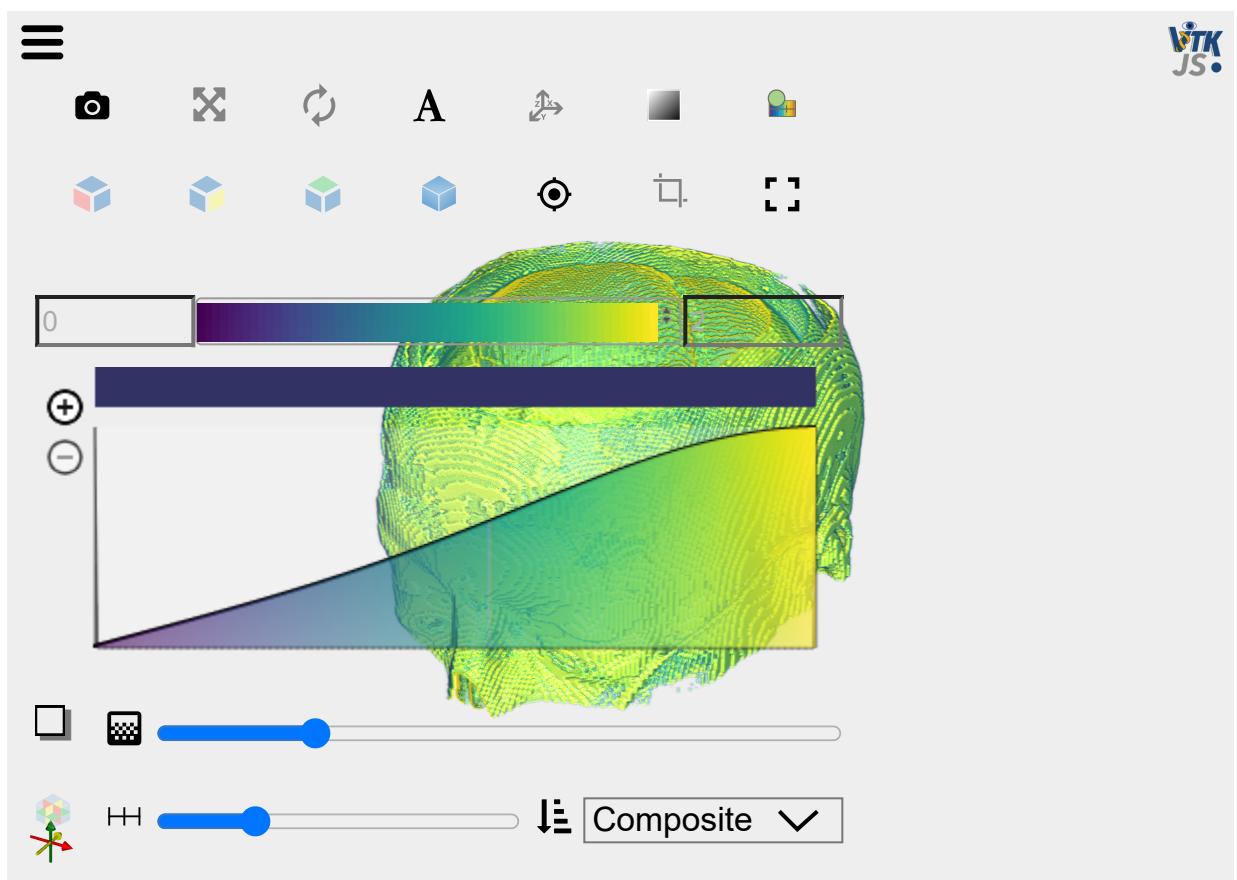
Again, there are better ways to segment brains from CTA. This method is very slow, but it works...

In [4]: `view(im1iso)`



In [5]: # Here we perform an initial segmentation of the skull using simple thresholding.  
# because of the standardization of CT to Hounsfield units.

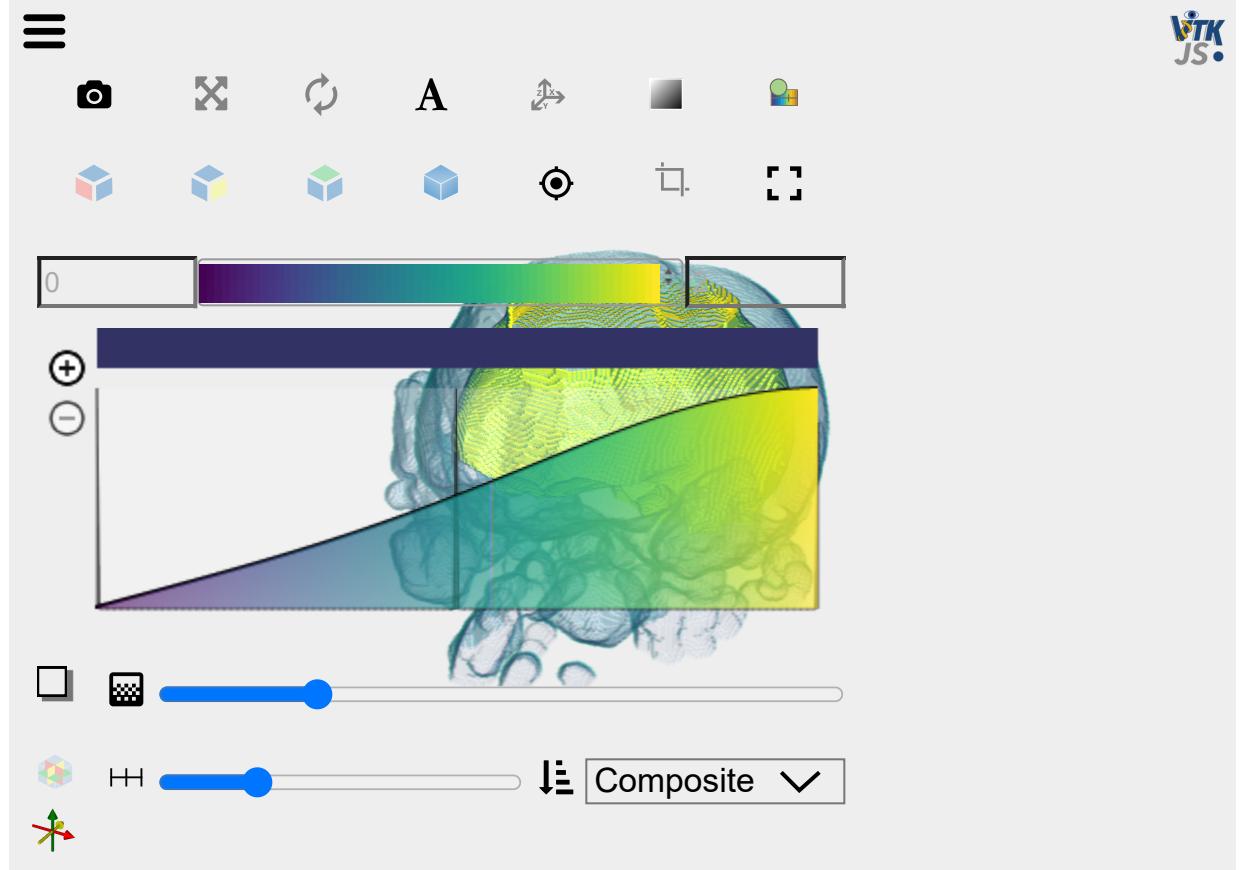
```
thresh = ttk.ImageMath.New(Input=im1iso)
thresh.ReplaceValuesOutsideMaskRange(im1iso, 1, 6000, 0)
thresh.ReplaceValuesOutsideMaskRange(im1iso, 0, 600, 1)
im1tmp = thresh.GetOutput()
thresh.ReplaceValuesOutsideMaskRange(im1tmp, 0, 1, 2)
im1Mask = thresh.GetOutputUChar()
view(im1Mask)
```



30

```
In [6]: # We then dilate and erode the mask to determine regions of high confidence of br  
# in the CTA  
  
maskMath = ttk.ImageMath.New(Input=im1Mask)  
maskMath.Threshold(0,1,0,1)  
maskMath.Erode(15,1,0)  
maskMath.Dilate(17,1,0)  
maskMath.Dilate(12,0,1)  
maskMath.Erode(12,1,0)  
brainSeed = maskMath.GetOutputUChar()  
maskMath.SetInput(im1Mask)  
maskMath.Threshold(2,2,0,1)  
maskMath.Erode(2,1,0)  
maskMath.Dilate(10,1,0)  
maskMath.Erode(7,1,0)  
skullSeed = maskMath.GetOutputUChar()  
maskMath.AddImages(brainSeed,1,2)  
comboSeed = maskMath.GetOutputUChar()
```

```
In [7]: view(comboSeed)
```



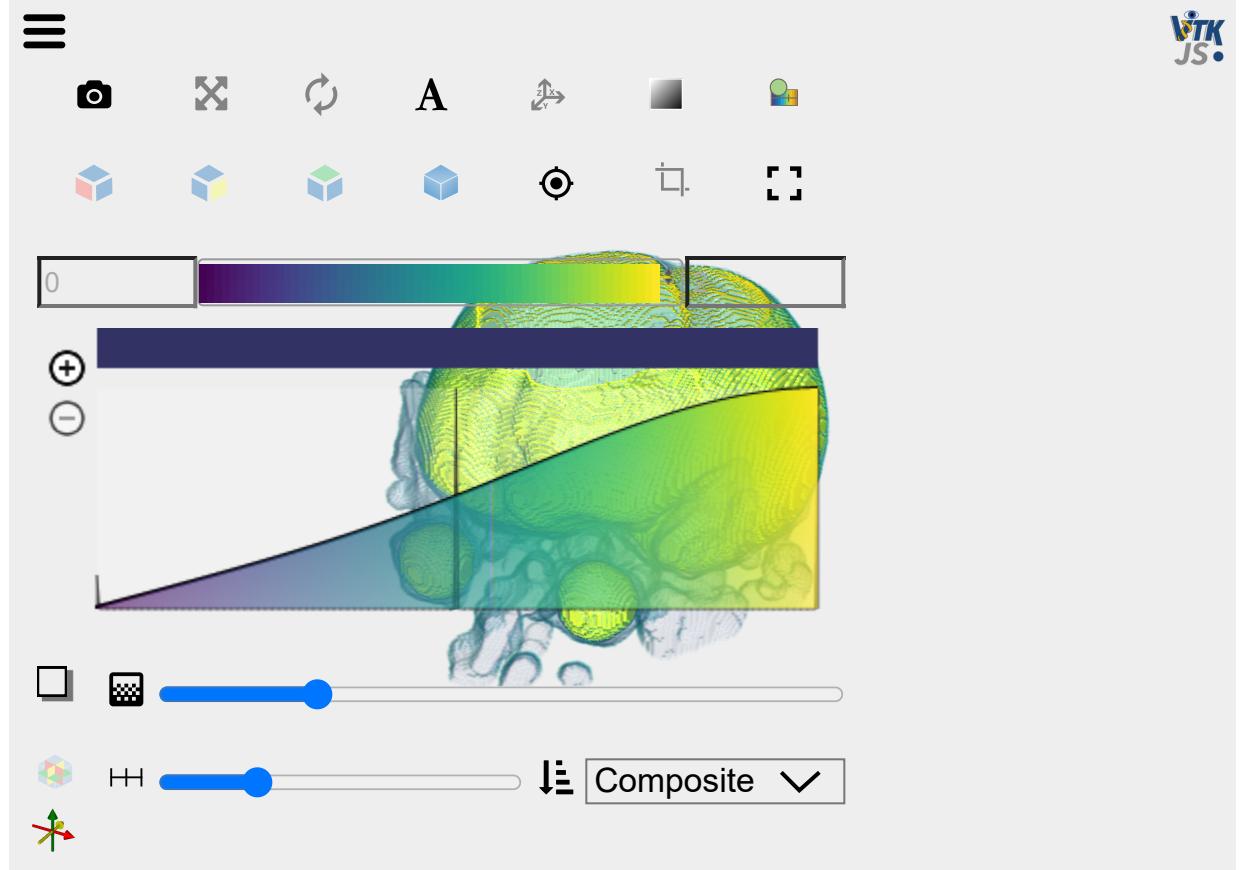
---

30

```
In [8]: # Finally, given the regions of high confidence, we run a probabilistic connected
# to determine the brain region of the CTA
LabelMapType = itk.Image[itk.UC,3]

segmenter = ttk.SegmentConnectedComponentsUsingParzenPDFs[ImageType,LabelMapType]
segmenter.SetFeatureImage( im1iso )
segmenter.SetInputLabelMap( comboSeed )
segmenter.SetObjectId( 2 )
segmenter.AddObjectId( 1 )
segmenter.SetVoidId( 0 )
segmenter.SetErodeDilateRadius( 20 )
segmenter.SetHoleFillIterations( 40 )
segmenter.Update()
segmenter.ClassifyImages()
brainMaskRaw = segmenter.GetOutputLabelMap()
```

```
In [9]: view(brainMaskRaw)
```



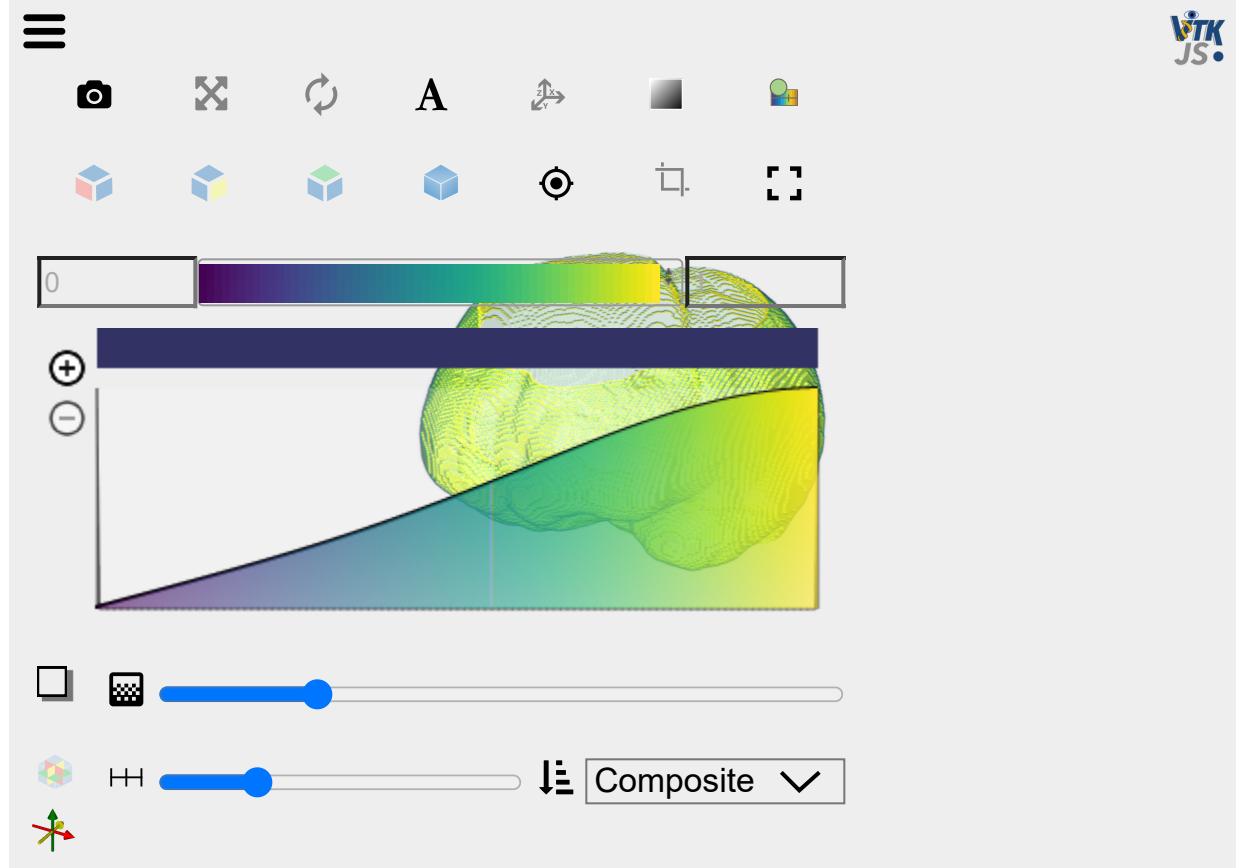
---

30

```
In [16]: maskMath.SetInput(brainMaskRaw)
maskMath.Threshold(2,2,1,0)
maskMath.Erode(1,1,0)
brainMaskRaw2 = maskMath.GetOutputUChar()

connComp = ttk.SegmentConnectedComponents.New(Input=brainMaskRaw2)
connComp.SetKeepOnlyLargestComponent(True)
connComp.Update()
brainMask = connComp.GetOutput()
```

In [17]: `view(brainMask)`



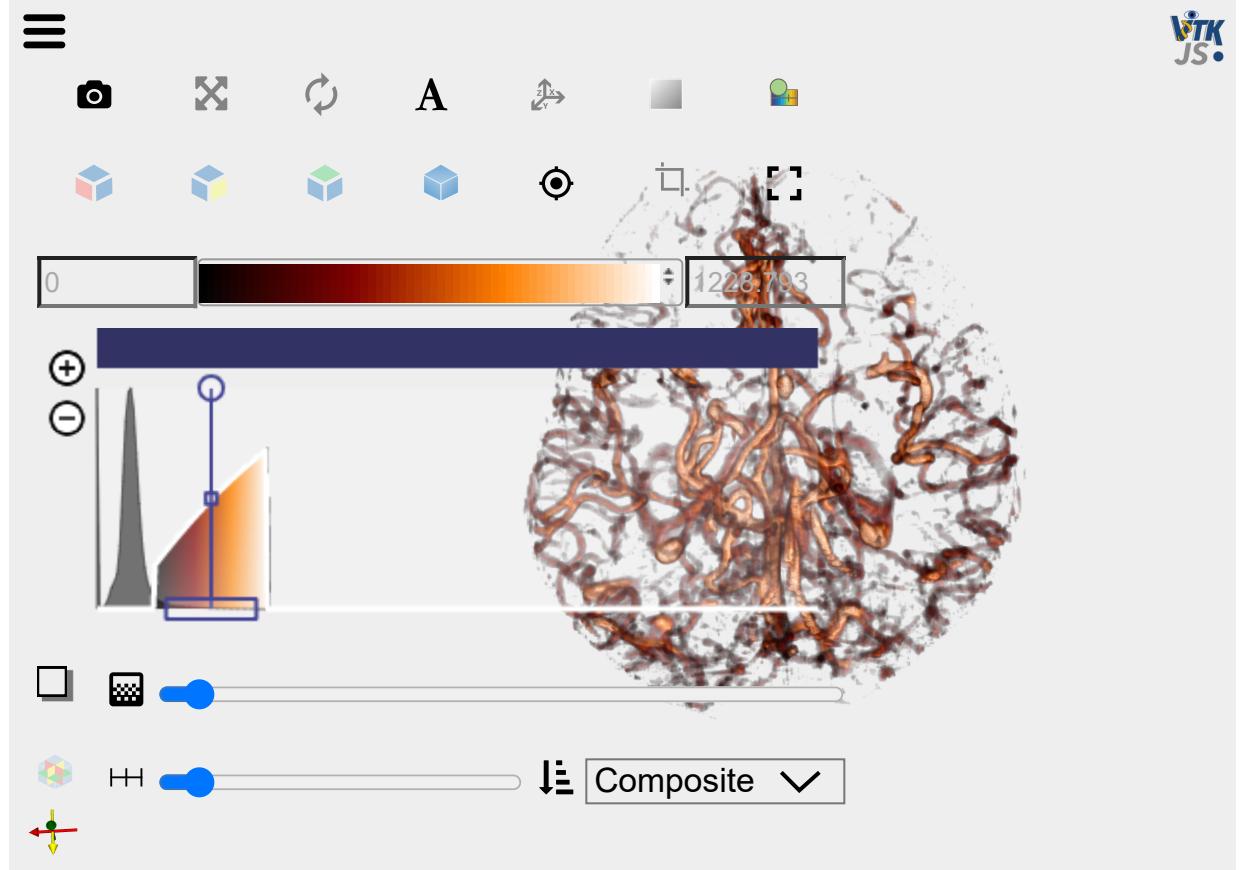
30

In [18]: # Then we mask the CTA - creating a CTA-Brain only image

```
cast = itk.CastImageFilter[LabelMapType, ImageType].New()
cast.SetInput(brainMask)
cast.Update()
brainMaskF = cast.GetOutput()

brainMath = ttk.ImageMath[ImageType, ImageType].New(Input = im1iso)
brainMath.ReplaceValuesOutsideMaskRange( brainMaskF, 1, 1, 0)
brain = brainMath.GetOutput()
```

In [19]: `view(brain)`



In [20]: `writer = itk.ImageFileWriter[ImageType].New(Input = brain)  
filename = InputBaseName + "-Brain.mha"  
writer.SetFileName(filename)  
writer.Update()`

In [ ]:

