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 demonstration how to "solve" brain extraction.

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

from itkwidgets import view

import numpy as np
```

In [2]:
```
InputBaseDir = "../Data/CTP-MinMax/"

CTPMaxFilename = InputBaseDir + "max3.mha"
CTPMinFilename = InputBaseDir + "min3.mha"
CTPBrainFilename = InputBaseDir + "max3-Brain.mha"

imMax = itk.imread(CTPMaxFilename, itk.F)
imMin = itk.imread(CTPMinFilename, itk.F)
imBrain = itk.imread(CTPBrainFilename, itk.F)
```

In [3]:
```
view(imBrain)
```

In [4]:
```
ImageType = itk.Image[itk.F, 3]

imMath = ttk.ImageMath.New(Input=imBrain)
imMath.Threshold( 0.00001, 2000, 1, 0)
imMath.Erode(10,1,0)
imBrainMaskErode = imMath.GetOutput()

imMath.SetInput(imMax)
imMath.AddImages(imMin,1,-1)
imDiff = imMath.GetOutput()
imMath.ReplaceValuesOutsideMaskRange(imBrain, 0.0001, 2000, 0)
imDiffBrain = imMath.GetOutput()
imMath.ReplaceValuesOutsideMaskRange(imBrainMaskErode, 0.5, 1.5, 0)
imDiffBrainErode = imMath.GetOutput()
```

In [5]:
```
tmpA = itk.GetArrayViewFromImage(imDiffBrain)
tmpAE = itk.GetArrayViewFromImage(imDiffBrainErode)
zMax = tmpA.shape[0]
clip = 0
while((np.amax(tmpA[clip:clip+1,:,:])>1000) | (np.amax(tmpA[clip:clip+1,:,:])==0)
    clip += 1
if(clip>0):
    tmpA[0:clip,:,:]=0
    tmpAE[0:clip,:,:]=0
clip = 1
while((np.amax(tmpA[zMax-clip:zMax-clip+1,:,:])>1000) | (np.amax(tmpA[zMax-clip:z
    clip += 1
print(clip, np.amax(tmpA[zMax-clip:zMax-clip+1,:,:]))
clip = clip - 1
if(clip>0):
    tmpA[zMax-clip:zMax,:,:]=0  #Happens to imDiffBrain since this array is a vie
    tmpAE[zMax-clip:zMax,:,:]=0  #Happens to imDiffBrain since this array is a vi
```

```
1 590.79504
```

In [6]:
```
view(imDiffBrain)
```

In [7]:
```
imMath = ttk.ImageMath[ImageType,ImageType].New()
imMath.SetInput(imDiffBrainErode)
imMath.Blur(1.5)
imBlur = imMath.GetOutput()
imBlurArray = itk.GetArrayViewFromImage(imBlur)

numSeeds = 15
seedCoverage = 20
seedCoord = np.zeros([numSeeds,3])
for i in range(numSeeds):
    seedCoord[i] = np.unravel_index(np.argmax(imBlurArray, axis=None), imBlurArra
    indx = [int(seedCoord[i][0]),int(seedCoord[i][1]),int(seedCoord[i][2])]
    minX = max(indx[0]-seedCoverage,0)
    maxX = max(indx[0]+seedCoverage,imBlurArray.shape[0])
    minY = max(indx[1]-seedCoverage,0)
    maxY = max(indx[1]+seedCoverage,imBlurArray.shape[1])
    minZ = max(indx[2]-seedCoverage,0)
    maxZ = max(indx[2]+seedCoverage,imBlurArray.shape[2])
    imBlurArray[minX:maxX,minY:maxY,minZ:maxZ]=0
    indx.reverse()
    seedCoord[:][i] = imDiffBrain.TransformIndexToPhysicalPoint(indx)
print(seedCoord)
```

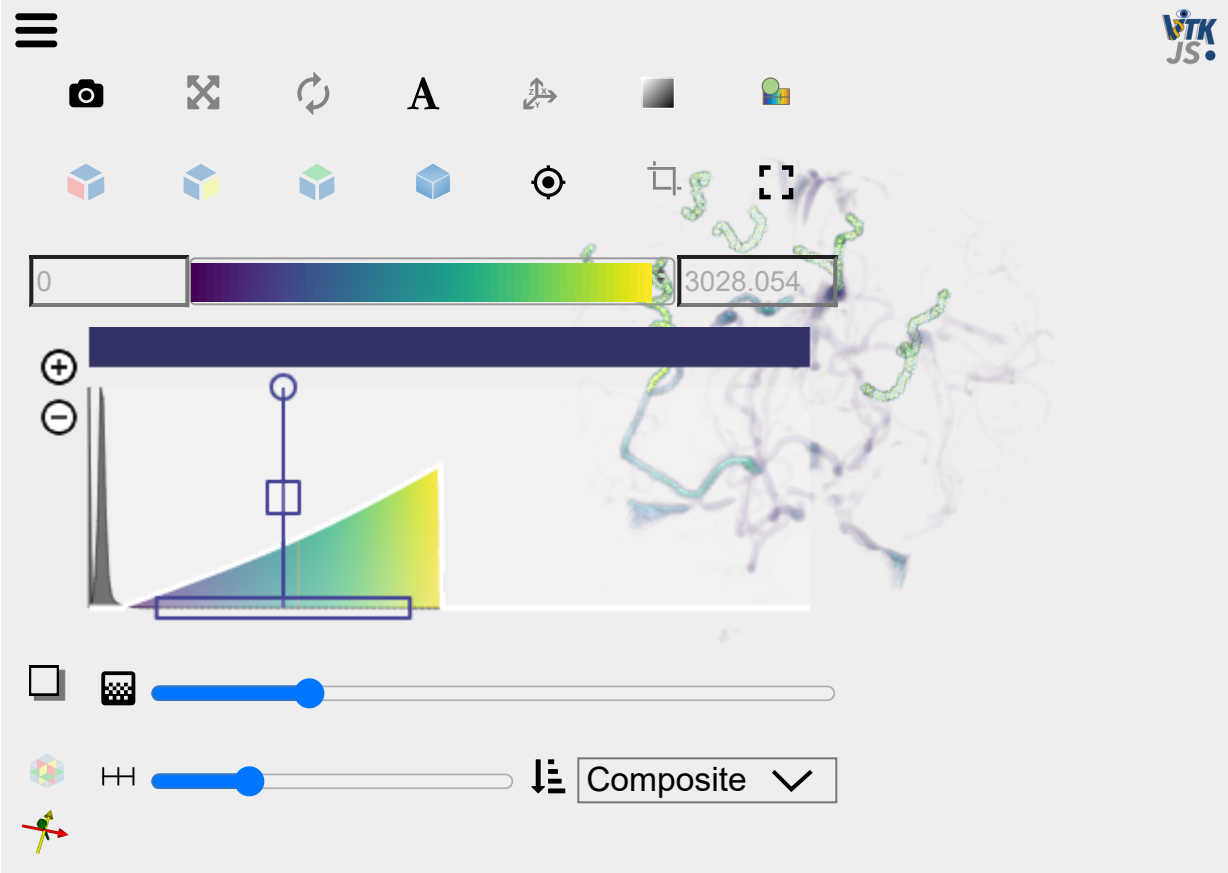
```
[[  -4.23602295 -145.46159363  296.73384094]
 [ -33.55116272 -160.39572144  298.94630432]
 [ -52.35710144 -206.85745239  296.73384094]
 [ -25.80754089 -144.90847778  278.48101807]
 [ -48.48529053 -193.02955627  305.03057861]
 [   0.74201965 -167.03311157  264.1000061 ]
 [ -45.16659546 -175.88296509  317.75224304]
 [  -5.34225464 -133.29304504  307.24304199]
 [ -58.44137573 -155.97079468  292.86203003]
 [  -6.44848633 -155.41767883  264.1000061 ]
 [ -43.50724792 -140.48355103  327.70832825]
 [  26.18534851 -133.29304504  285.1184082 ]
 [ -46.82594299 -157.63014221  276.82167053]
 [ -18.61703491 -128.86811829  290.64956665]
 [ -31.33869934 -128.86811829  297.28695679]]
```

In [8]:
```
# Manually extract a few vessels to form an image-specific training set
vSeg = ttk.SegmentTubes.New(Input=imDiffBrain)
vSeg.SetVerbose(True)
vSeg.SetMinRoundness(0.4)
vSeg.SetMinCurvature(0.002)
vSeg.SetRadiusInObjectSpace( 1 )
for i in range(numSeeds):
    print("**** Processing seed " + str(i) + " : " + str(seedCoord[i]))
    vSeg.ExtractTubeInObjectSpace( seedCoord[i], i )

tubeMaskImage = vSeg.GetTubeMaskImage()
```

```
**** Processing seed 0 : [  -4.23602295 -145.46159363  296.73384094]
**** Processing seed 1 : [ -33.55116272 -160.39572144  298.94630432]
**** Processing seed 2 : [ -52.35710144 -206.85745239  296.73384094]
**** Processing seed 3 : [ -25.80754089 -144.90847778  278.48101807]
**** Processing seed 4 : [ -48.48529053 -193.02955627  305.03057861]
**** Processing seed 5 : [   0.74201965 -167.03311157  264.1000061 ]
**** Processing seed 6 : [ -45.16659546 -175.88296509  317.75224304]
**** Processing seed 7 : [  -5.34225464 -133.29304504  307.24304199]
**** Processing seed 8 : [ -58.44137573 -155.97079468  292.86203003]
**** Processing seed 9 : [  -6.44848633 -155.41767883  264.1000061 ]
**** Processing seed 10 : [ -43.50724792 -140.48355103  327.70832825]
**** Processing seed 11 : [  26.18534851 -133.29304504  285.1184082 ]
**** Processing seed 12 : [ -46.82594299 -157.63014221  276.82167053]
**** Processing seed 13 : [ -18.61703491 -128.86811829  290.64956665]
**** Processing seed 14 : [ -31.33869934 -128.86811829  297.28695679]
```

In [15]:
```
imMath.SetInput(tubeMaskImage)
imMath.AddImages(imDiffBrain, 200, 1)
blendIm = imMath.GetOutput()
view(blendIm)
```

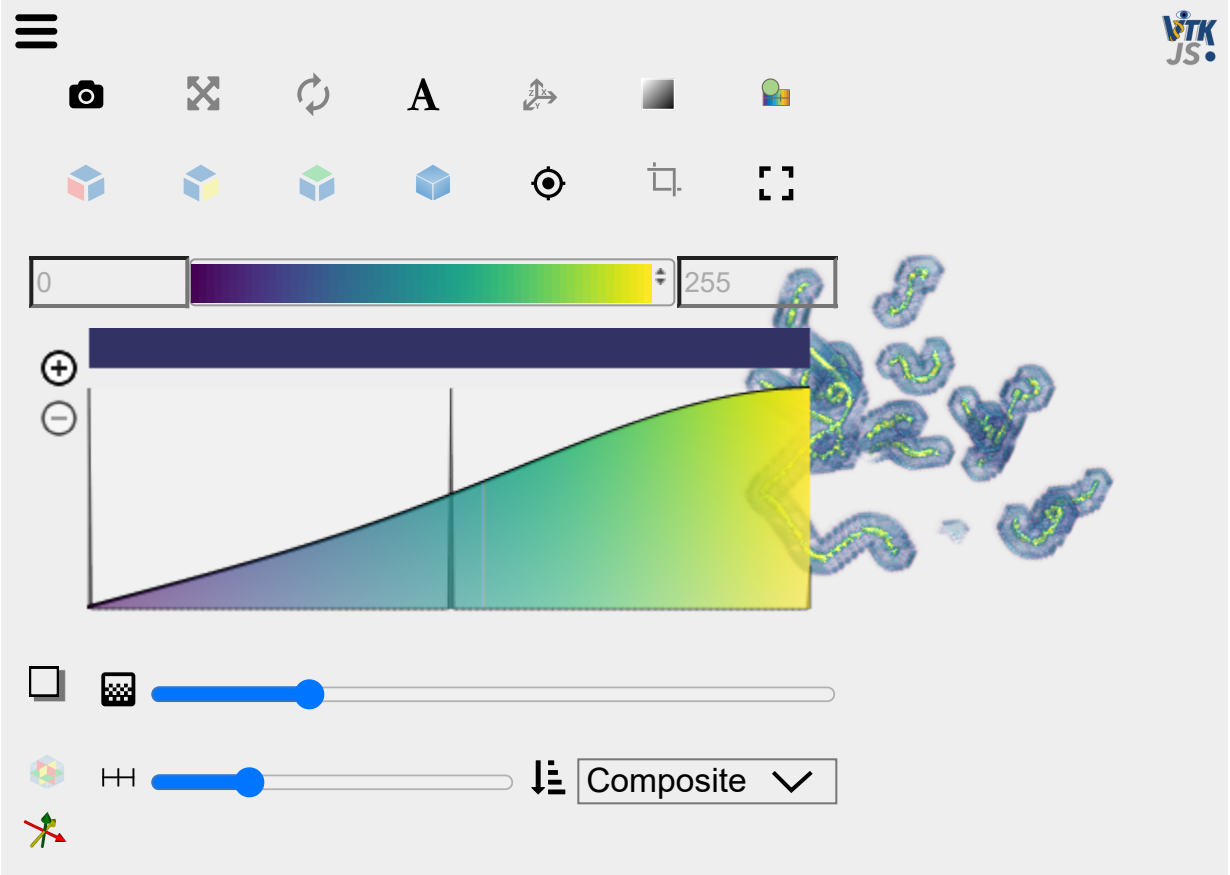


20

In [16]:
```
LabelMapType = itk.Image[itk.UC,3]

trMask = ttk.ComputeTrainingMask[ImageType,LabelMapType].New()
trMask.SetInput( tubeMaskImage )
trMask.SetGap( 4 )
trMask.SetObjectWidth( 1 )
trMask.SetNotObjectWidth( 1 )
trMask.Update()
fgMask = trMask.GetOutput()
```

In [17]:
```
view(fgMask)
```



20

In [*]:
```
enhancer = ttk.EnhanceTubesUsingDiscriminantAnalysis[ImageType,LabelMapType].New(
enhancer.AddInput( imDiff )
enhancer.SetLabelMap( fgMask )
enhancer.SetRidgeId( 255 )
enhancer.SetBackgroundId( 128 )
enhancer.SetUnknownId( 0 )
enhancer.SetTrainClassifier(True)
enhancer.SetUseIntensityOnly(True)
enhancer.SetScales([0.43,1.29,3.01])
enhancer.Update()
enhancer.ClassifyImages()
```

In [*]:
```
im1vess = itk.SubtractImageFilter( Input1=enhancer.GetClassProbabilityImage(0), I

imMath.SetInput(imDiffBrain)
imMath.Threshold(0.0001,2000,1,0)
imMath.Erode(2,1,0)
imBrainE = imMath.GetOutput()

imMath.SetInput(im1vess)
imMath.ReplaceValuesOutsideMaskRange(imBrainE, 1, 1, -0.001)
im1vessBrain = imMath.GetOutput()
#view(enhancer.GetClassProbabilityImage(0))
view(im1vessBrain)
```

In [*]:
```
itk.imwrite( im1vess, InputBaseDir + "diff3-VesselEnhanced.mha", compression=True

itk.imwrite( im1vessBrain, InputBaseDir + "diff3-Brain-VesselEnhanced.mha", compr
```

In [ ]:
```
```