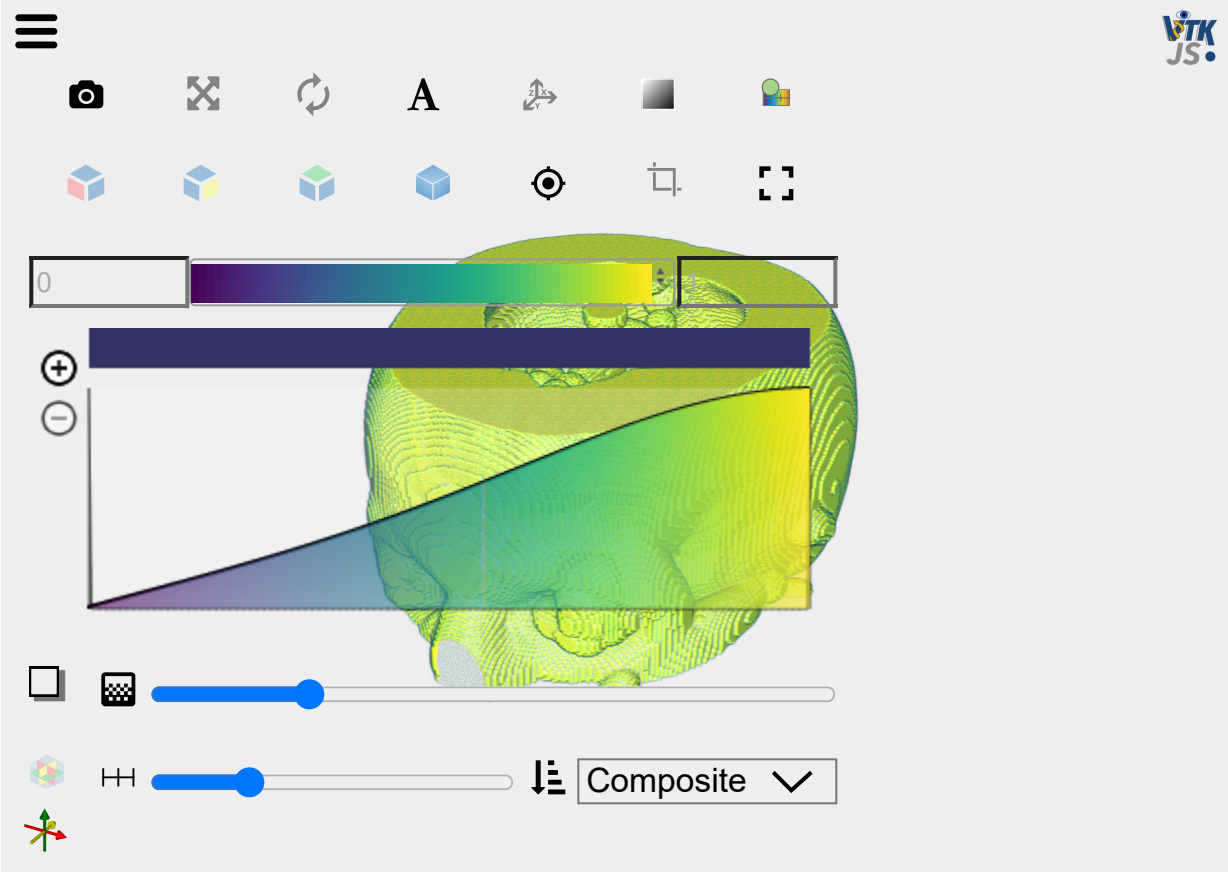
In [1]:
```
import os
import sys
import glob
import numpy as np
import time

import itk
from itk import TubeTK as ttk
from itkwidgets import view
```

In [2]:
```
# NRRD Study Name
studyname = '../Data/CTP'

# NRRD Files
directory = (studyname + '/')

# Saved NRRD Files
directory2 = (studyname + '-Reg/')
if os.path.isdir(directory2) == False:
    os.mkdir(directory2)

# Mask Creation and Location
directory3 = (studyname + '-MinMax/')
if os.path.isdir(directory3) == False:
    os.mkdir(directory3)

pic_folder = os.listdir(directory)
pic_folder = [pic_folder for pic_folder in pic_folder if ".mha" in pic_folder]
pic_folder.sort()
print(pic_folder)
num_images = len(pic_folder)

im0Tmp = itk.imread(directory + pic_folder[int(num_images/2)], itk.F)

resample = ttk.ResampleImage.New(Input=im0Tmp,MakeIsotropic=True)
resample.Update()
im0 = resample.GetOutput()
immath = ttk.ImageMath.New(Input=im0)
immath.Blur(1)
im0Blur = immath.GetOutput()

immath.Threshold(150, 800, 1, 0)
immath.Dilate(10, 1, 0)
mask0 = immath.GetOutputUChar()
mask0Tmp = itk.GetArrayViewFromImage(mask0)
mask0Tmp[0:4,:,:] = 0
sizeZ = mask0Tmp.shape[0]
mask0Tmp[sizeZ-4:sizeZ,:,:] = 0   #No need to update mask0 since mask0Tmp is a vi

itk.imwrite(mask0, directory3 + 'mask.mha', compression=True)
maskObj = itk.ImageMaskSpatialObject[3].New()
maskObj.SetImage(mask0)
maskObj.Update()
```

```
['CTP04.mha', 'CTP06.mha', 'CTP08.mha', 'CTP10.mha', 'CTP12.mha', 'CTP14.mha',
'CTP16.mha', 'CTP18.mha', 'CTP20.mha', 'CTP22.mha', 'CTP24.mha', 'CTP26.mha',
'CTP28.mha', 'CTP30.mha', 'CTP32.mha']
```

In [3]:
```
view(mask0)
```



30

In [4]:
```
Dimension = 3
PixelType = itk.ctype('float')
ImageType = itk.Image[PixelType, Dimension]

imdatamax = itk.GetArrayFromImage(im0)
imdatamin = imdatamax
imdatamax2 = imdatamax
imdatamin2 = imdatamax
imdatamax3 = imdatamax
imdatamin3 = imdatamax

imFixedBlur = im0Blur

for imNum in range(num_images):
    start = time.time()

    imMoving = itk.imread( directory + pic_folder[imNum], itk.F )

    immath.SetInput(imMoving)
    immath.Blur(1)
    imMovingBlur = immath.GetOutput()

    imreg = ttk.RegisterImages[ImageType].New()
    imreg.SetFixedImage(imFixedBlur)
    imreg.SetMovingImage(imMovingBlur)

    imreg.SetRigidMaxIterations(3000)
    imreg.SetRegistration("RIGID")
    imreg.SetExpectedOffsetMagnitude(20)
    imreg.SetExpectedRotationMagnitude(0.3)
    imreg.SetMetric("MEAN_SQUARED_ERROR_METRIC")

    imreg.SetFixedImageMaskObject(maskObj)

    #imreg.SetReportProgress(True)

    imreg.Update()

    tfm = imreg.GetCurrentMatrixTransform()
    imMovingReg = imreg.ResampleImage("LINEAR_INTERPOLATION", imMoving, tfm, -102

    itk.imwrite( imMovingReg, directory2 + pic_folder[imNum], compression=True )

    imdataTmp = itk.GetArrayFromImage(imMovingReg)

    imdatamax = np.maximum(imdatamax,imdataTmp)
    imdatamin = np.minimum(imdatamin,imdataTmp)
    imdataTmp[np.where(imdataTmp==imdatamax)] = 0
    imdataTmp[np.where(imdataTmp==imdatamin)] = 0
    imdatamax2 = np.maximum(imdatamax2,imdataTmp)
    imdatamin2 = np.minimum(imdatamin2,imdataTmp)
    imdataTmp[np.where(imdataTmp==imdatamax)] = 0
    imdataTmp[np.where(imdataTmp==imdatamin)] = 0
    imdatamax3 = np.maximum(imdatamax3,imdataTmp)
    imdatamin3 = np.minimum(imdatamin3,imdataTmp)
```

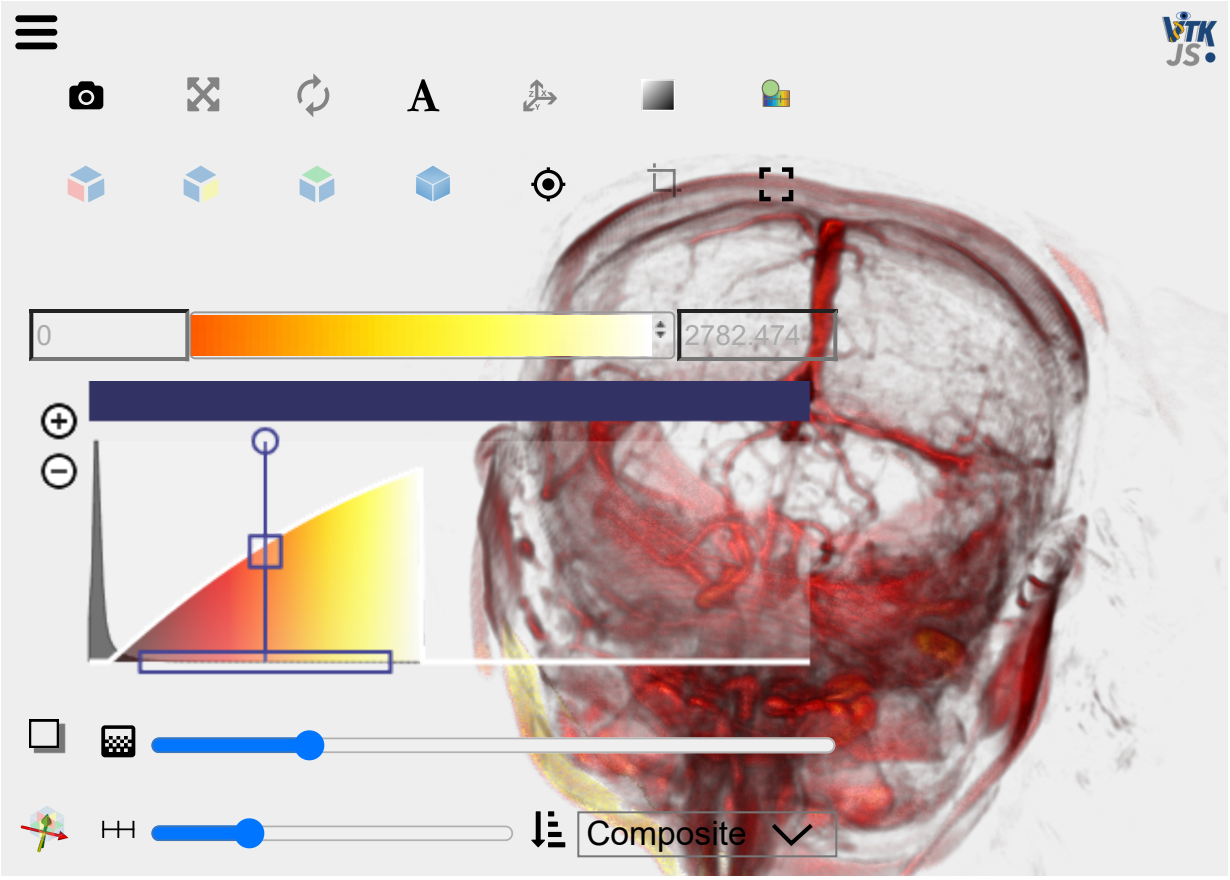
```
    end = time.time()

    percent = (imNum + 1) / num_images * 100
    print('***  ' + str(round(percent)) + '% : ' + str(round(end-start)) + 's : '

print('Done')
```

```
***  7% : 59s : CTP04.mha  ***
***  13% : 65s : CTP06.mha  ***
***  20% : 63s : CTP08.mha  ***
***  27% : 63s : CTP10.mha  ***
***  33% : 63s : CTP12.mha  ***
***  40% : 62s : CTP14.mha  ***
***  47% : 70s : CTP16.mha  ***
***  53% : 67s : CTP18.mha  ***
***  60% : 67s : CTP20.mha  ***
***  67% : 65s : CTP22.mha  ***
***  73% : 68s : CTP24.mha  ***
***  80% : 69s : CTP26.mha  ***
***  87% : 70s : CTP28.mha  ***
***  93% : 68s : CTP30.mha  ***
***  100% : 68s : CTP32.mha  ***
Done
```

In [5]:
```
out = itk.GetImageFromArray(imdatamax3)
out.CopyInformation(im0)
itk.imwrite(out, (directory3 + 'max3.mha'), compression=True)

out = itk.GetImageFromArray(imdatamin3)
out.CopyInformation(im0)
itk.imwrite(out, (directory3 + 'min3.mha'), compression=True)

out = itk.GetImageFromArray(imdatamax3 - imdatamin3)
out.CopyInformation(im0)
itk.imwrite(out, (directory3 + 'diff3.mha'), compression=True)
```

In [6]:
```
out = itk.GetImageFromArray(imdatamax)
out.CopyInformation(im0)
itk.imwrite(out, (directory3 + 'max.mha'), compression=True)

out = itk.GetImageFromArray(imdatamin)
out.CopyInformation(im0)
itk.imwrite(out, (directory3 + 'min.mha'), compression=True)

out = itk.GetImageFromArray(imdatamax - imdatamin)
out.CopyInformation(im0)
itk.imwrite(out, (directory3 + 'diff.mha'), compression=True)
```

In [7]:
```
out = itk.GetImageFromArray(imdatamax-imdatamin)
out.CopyInformation(im0)
view(out)
```



30

In [ ]: