In [1]:
```
import cv2
from scipy import ndimage
from skimage import io, color, measure, exposure, img_as_ubyte
import numpy as np
import h5py
import hdf5plugin
import os
import matplotlib.pyplot as plt
import napari
import pyclesperanto_prototype as cle   #https://github.com/clEsperanto/pyclespe
import skimage.measure as skm
from scipy.spatial import Voronoi
from scipy.ndimage import zoom
```

In [2]:
```
h5name = 'C:/Users/AO0736/Box/Indrajeet/LabDCT/AluFoams_3p2mm_bigscan_30s_export

with h5py.File(h5name, "r") as hin:
    tomo = hin['AbsorptionCT']['Data'][:]
    labdct = hin['LabDCT']['Data']['GrainId'][:]*hin['LabDCT']['Data']['Mask'][:
```

In [3]:
```
labdct.shape
```

Out[3]:
```
(310, 332, 333)
```

In [4]:
```
zoom_factors = [c/g for c,g in zip(labdct.shape,tomo.shape)]
print(zoom_factors)
tomo_compress=zoom(tomo,zoom_factors,order=0)
print(labdct.shape)
print(tomo_compress.shape)
```

```
[0.33769063180827885, 0.3377416073245168, 0.33772819472616633]
(310, 332, 333)
(310, 332, 333)
```

In [5]:
```
import pandas as pd
props = skm.regionprops(labdct)
pd.DataFrame(props)
```

Out[5]:

	0	1	2	3	4	5	6
0	area	area_bbox	area_convex	area_filled	axis_major_length	axis_minor_length	bbox
1	area	area_bbox	area_convex	area_filled	axis_major_length	axis_minor_length	bbox
2	area	area_bbox	area_convex	area_filled	axis_major_length	axis_minor_length	bbox
3	area	area_bbox	area_convex	area_filled	axis_major_length	axis_minor_length	bbox
4	area	area_bbox	area_convex	area_filled	axis_major_length	axis_minor_length	bbox
...	...	...	...	...	...	...	...
130	area	area_bbox	area_convex	area_filled	axis_major_length	axis_minor_length	bbox
131	area	area_bbox	area_convex	area_filled	axis_major_length	axis_minor_length	bbox
132	area	area_bbox	area_convex	area_filled	axis_major_length	axis_minor_length	bbox
133	area	area_bbox	area_convex	area_filled	axis_major_length	axis_minor_length	bbox
134	area	area_bbox	area_convex	area_filled	axis_major_length	axis_minor_length	bbox

135 rows × 30 columns

In [6]:
```
points = np.zeros((3,135))

for i in range (134):
    x_pos = props[i]['centroid'][0]
    x_pos = np.asarray(x_pos)

    y_pos = props[i]['centroid'][1]
    y_pos = np.asarray(y_pos)

    z_pos = props[i]['centroid'][2]
    z_pos = np.asarray(z_pos)

    points[0,i]= x_pos
    points[1,i]= y_pos
    points[2,i]= z_pos

    #print(a,b,c)

print(points.shape)
```

```
(3, 135)
```

In [9]:
```
# selet a GPU with the following in the name. This will fallback to any other GP
cle.select_device('RTX')

image = cle.create((310, 332, 333))
cle.set(image, 0)

cle.pointlist_to_labelled_spots(points, image)

spots = cle.maximum_sphere(image, radius_x=10, radius_y=10, radius_z=10)
```

In [10]:
```
spots_image.shape
```

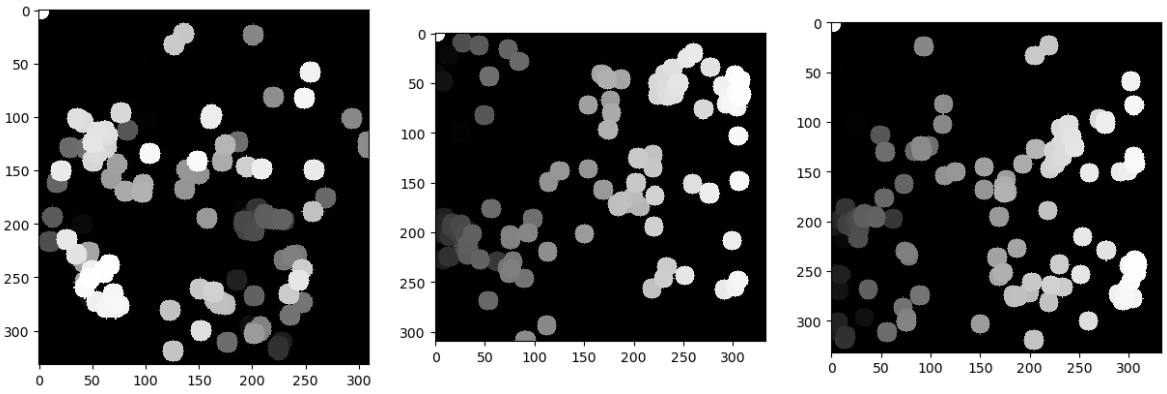
Out[10]: (310, 332, 333)

In [12]:
```
def show(image_to_show, labels=False):
    """
    This function generates three projections: in X-, Y- and Z-direction and sho
    """
    projection_x = cle.maximum_x_projection(image_to_show)
    projection_y = cle.maximum_y_projection(image_to_show)
    projection_z = cle.maximum_z_projection(image_to_show)

    fig, axs = plt.subplots(1, 3, figsize=(15, 15))
    cle.imshow(projection_x, plot=axs[0], labels=labels)
    cle.imshow(projection_y, plot=axs[1], labels=labels)
    cle.imshow(projection_z, plot=axs[2], labels=labels)

show(spots)
print(spots.shape)
```

(310, 332, 333)



In [23]:
```
spots_image = cle.pull(spots)
```

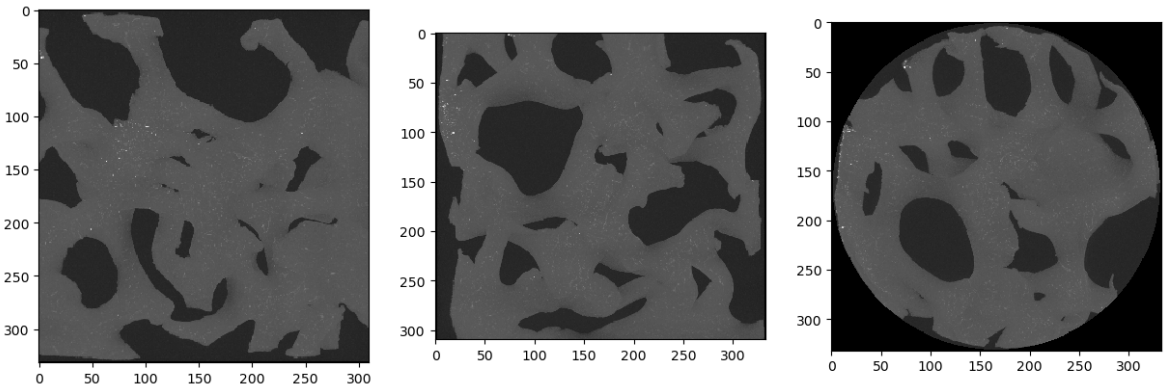
In [28]:
```
viewer = napari.view_image(tomo_compress)
viewer = napari.view_image(spots_image)
```

In [ ]:
```
```

In [27]:
```
#lets look at how this data look

show(tomo_compress)
print(tomo_compress.shape)
```

(310, 332, 333)



In [ ]: