ALS User Meeting 2022

This notebook describes methods to extract key information from microCT image stacks

- Thresholding techniques
- Image enhancing techniques
- Differential operators for edge detection

Created by Dani Ushizima, CAMERA, LBNL - Aug 1st 2022

```
%matplotlib inline
```

```
import numpy as np
from scipy import ndimage as ndi
import fnmatch,os
import matplotlib.pyplot as plt
from glob import glob

from skimage import img_as_ubyte, filters, morphology, exposure, io, restoration
from skimage.filters import threshold_isodata
from skimage.transform import pyramid_expand
from skimage.measure import regionprops,label
```

1. Read a microct image

- from url
- from NERSC
- from Google drive

Read from NERSC

- discard this portion if running in Colab

```
datapath = "/global/cfs/cdirs/als/users/yourname/yourdata/" #update these values
!ls -lt "$datapath"
```

```
image = io.imread(datapath+'bead_pack.tif')
```

Read from Google drive

- discard this portion if running at NERSC

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
datapath = "/content/drive/My Drive/Colab Notebooks/ALS User Meeting 2022 colab/data/'
!ls -lt "$datapath"
```

```
total 7842
-rw------- 1 root root 8025493 Aug 11 16:46 bead_pack.tif
drwx------ 2 root root    4096 Aug 11 16:35 concrete
```

```
def loadFileNames(path,extension):
  ''' Return filename after using colab files.upload - work for 1 file'''
  fnames = glob(path+extension)
  fnames.sort()
  print(path);
  print(f"Number of files: {len(fnames)}")
  return fnames
```

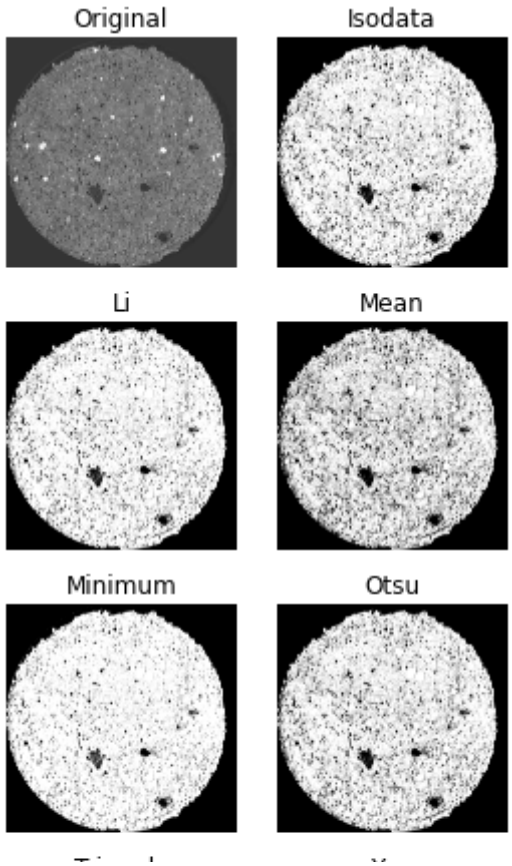
```
extension = '*tif'
files = loadFileNames(datapath+'concrete/',extension)
```

```
/content/drive/My Drive/Colab Notebooks/ALS User Meeting 2022 colab/data/concret
Number of files: 20
```

2. Thresholding techniques

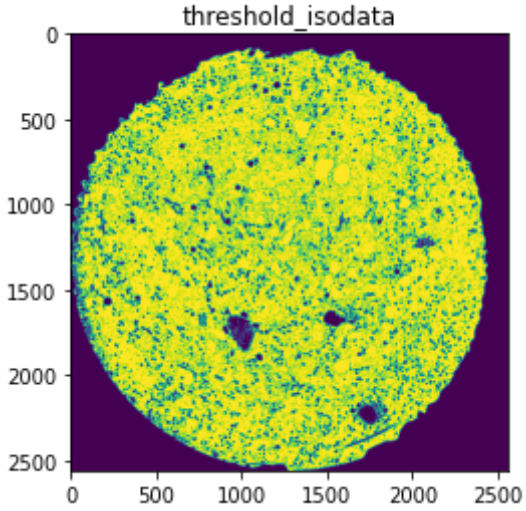
- use a slice to test many approaches

```
n = 5
aslice = io.imread(files[n])
t= filters.try_all_threshold(aslice, figsize=(4,8), verbose=False)
#plt.savefig("thresholdAll.png", bbox_inches='tight')
```



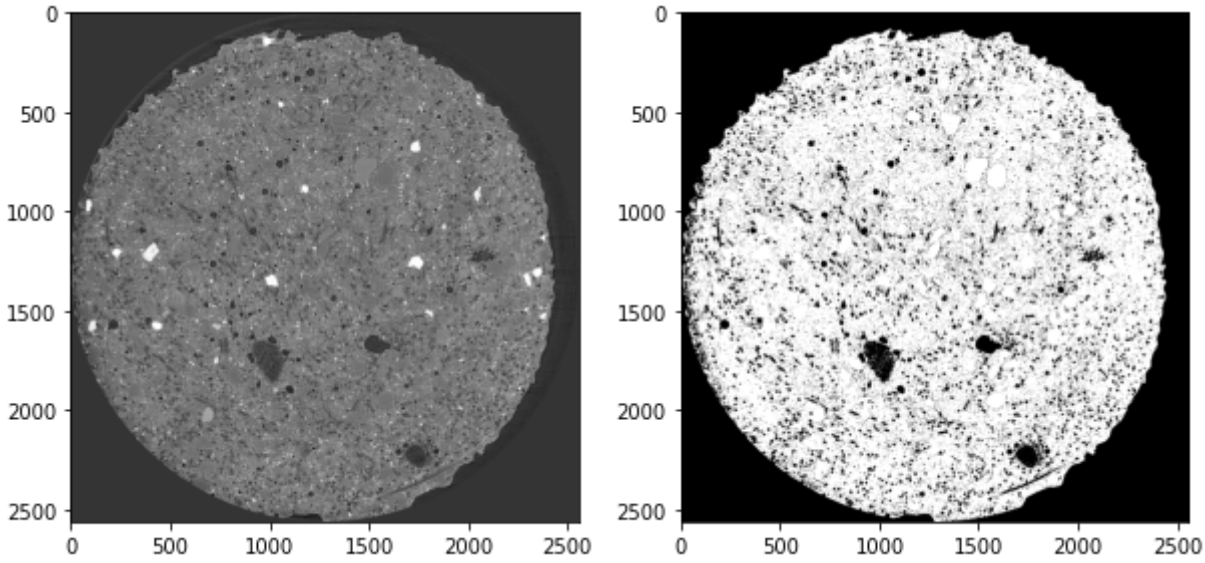

```
t=filters.threshold_isodata(aslice) #calculates the threshold
binary = aslice>t #apply to image
plt.imshow(binary)
plt.title('threshold_isodata')
```

```
Text(0.5, 1.0, 'threshold_isodata')
```




```
def imshowcmp(before,after,lut='plasma'):
    '''Show 2 images side by side'''
    f, ax = plt.subplots(1, 2, figsize=(10, 10))
    ax[0].imshow(before,cmap=lut)
    ax[1].imshow(after,cmap=lut)
```

```
imshowcmp(aslice,binary,'gray')
```

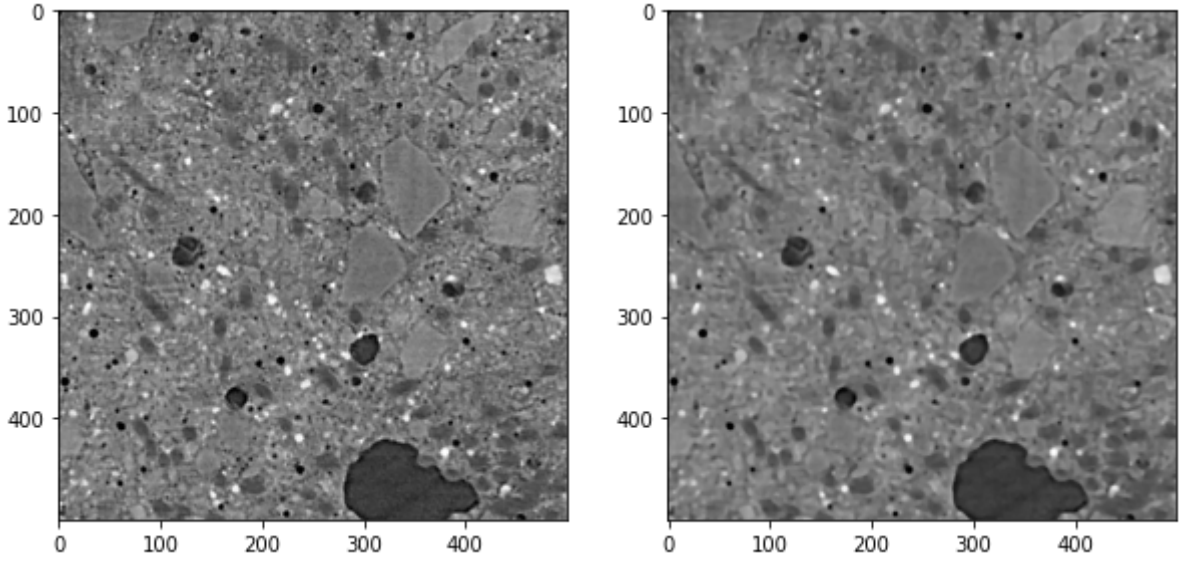


3. Image enhancing techniques

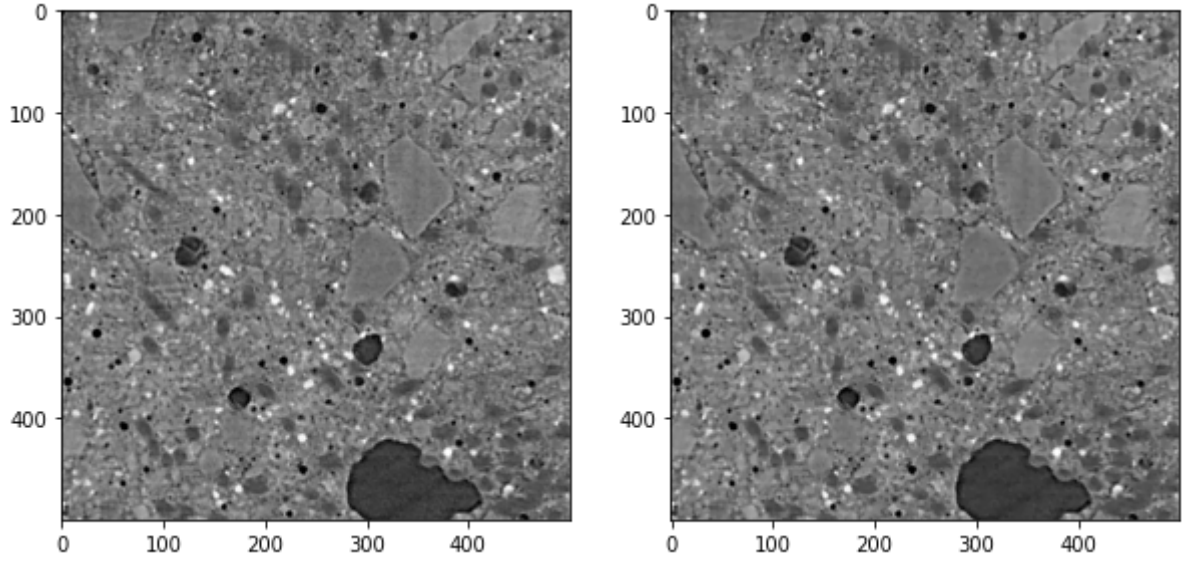
- denoising: median, bilateral
- sharpening
- morphological operators

```
aslice = aslice[1200:1700,1200:1700]
```

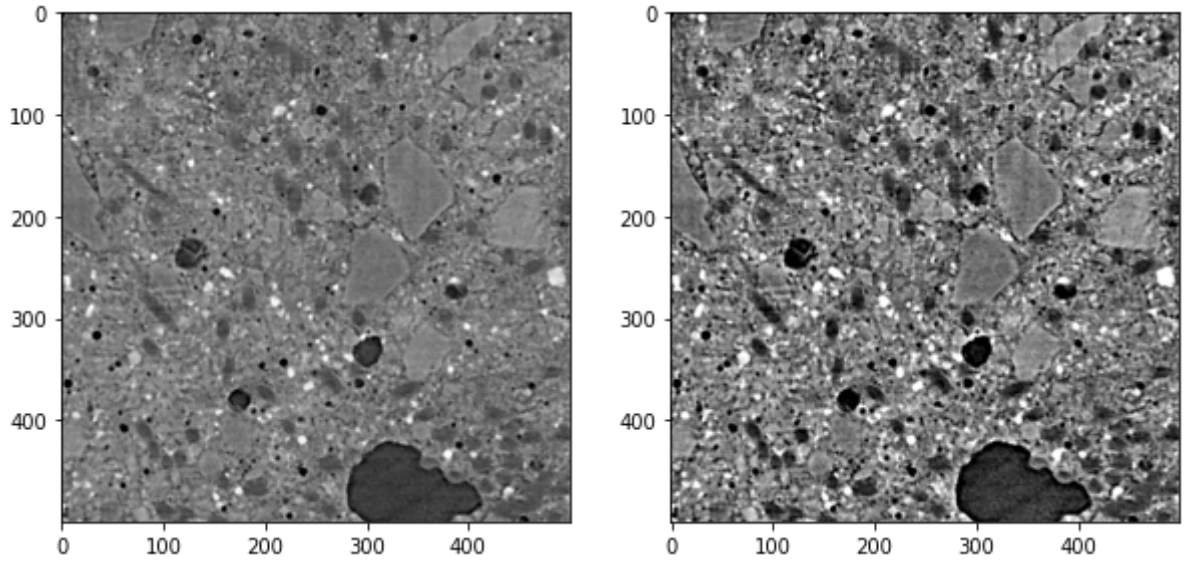
```
selem = morphology.diamond(3)
medianImage = filters.median(aslice,selem)
imshowcmp(aslice,medianImage,'gray')
```



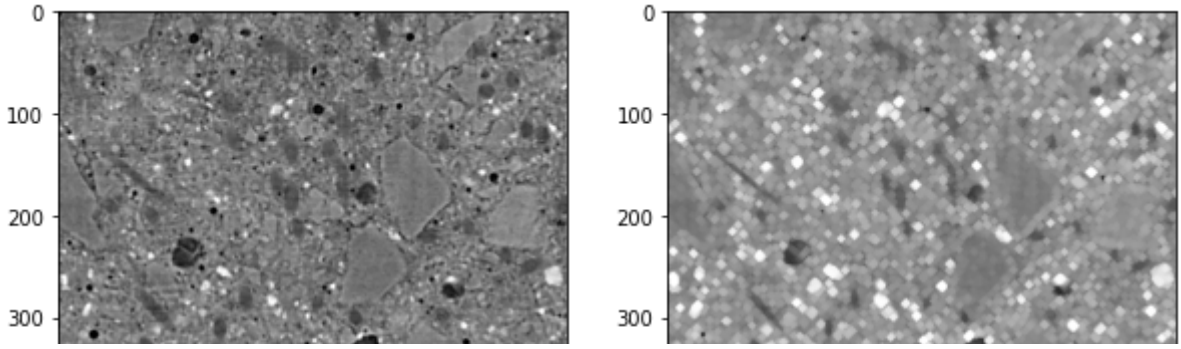
```
#from skimage import restoration
bilatImage = filters.rank.mean_bilateral(aslice,selem)
imshowcmp(aslice,bilatImage,'gray')
```



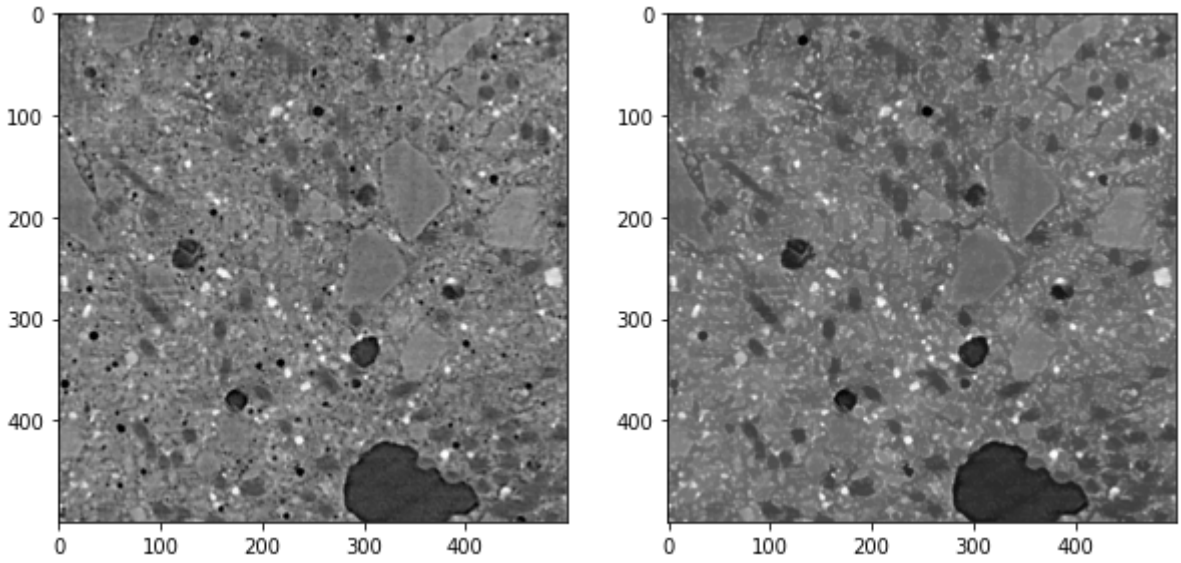

```
sharpImage = filters.unsharp_mask(aslice, radius=20, amount=1)
imshowcmp(aslice,sharpImage,'gray')
```




```
dilatImage = morphology.dilation(aslice,selem=selem)
imshowcmp(aslice,dilatImage,'gray')
```



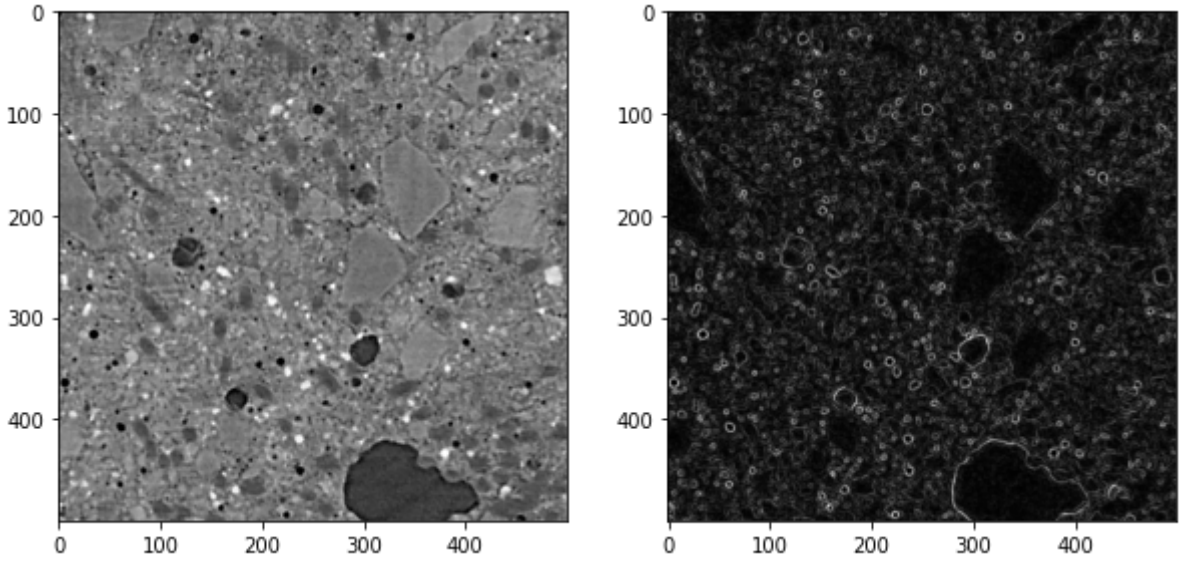
```
dilatImage = morphology.area_closing(aslice)
imshowcmp(aslice,dilatImage,'gray')
```



4. Differential operators for edge detection

- sobel
- other options

```
sobelImage = filters.sobel(aslice)
imshowcmp(aslice,sobelImage,'gray')
```



✓ 0s completed at 2:55 PM