# -\*- coding: utf-8 -\*-

import numpy as np

import matplotlib.pyplot as plt

from skimage.io import imread

from skimage import data\_dir

from skimage.transform import radon

from skimage.transform import iradon

 #画像のインポート

image = imread( "C:\\Users\\●●●\\●●.png", as\_grey=True)

imgNum=max(image.shape)

thetaNum=500

theta = np.linspace(0., 180., thetaNum, endpoint=False)

 #ラドン変換

sinogram = radon(image, theta=theta, circle=True)

 #元画像とサイノグラムの表示

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8, 4.5))

ax1.set\_title("Original")

ax1.imshow(image, cmap=plt.cm.Greys)

ax2.set\_title("Radon transform\n(Sinogram)")

ax2.set\_xlabel("Projection angle (deg)")

ax2.set\_ylabel("Projection position (pixels)")

ax2.imshow(sinogram, cmap=plt.cm.hot,

 extent=(0, 180, 0, sinogram.shape[0]), aspect='auto')

fig.tight\_layout()

plt.show()

#FBPによる画像の復元(filterのデフォルトはランプフィルタ. filter=Noneでフィルタなしに)

reconstruction\_fbp = iradon(sinogram, theta=theta, circle=True)

#復元画像と元の画像の誤差

#error = reconstruction\_fbp - image

#平均二乗誤差

#print('FBP rms reconstruction error: %.3g' % np.sqrt(np.mean(error\*\*2)))

 #結果表示

imkwargs = dict(vmin=-0.2, vmax=0.2)

fig2, (ax3, ax4) = plt.subplots(1, 2, figsize=(8, 4.5),

 sharex=True, sharey=True,

 subplot\_kw={'adjustable': 'box-forced'})

ax3.set\_title("Reconstruction\nFiltered back projection")

ax3.imshow(reconstruction\_fbp, cmap=plt.cm.Greys)

#ax4.set\_title("Reconstruction error\nFiltered back projection")

#ax4.imshow(reconstruction\_fbp - image, cmap=plt.cm.Greys\_r, \*\*imkwargs)

plt.show()