Filtro_maestro

October 25, 2019

1 División de Ciencias e Ingenierías de la Universidad de Guanajuato

1.1 Fundamentos de procesamiento digital de imágenes

1.2 TAREA : Funciones de filtrado en frecuencia

1.2.1 Profesor : Dr. Arturo González Vega

1.2.2 Alumno : Gustavo Magaña López

[1]:
```
import copy
import importlib
from typing import Tuple, List, NoReturn

import numpy as np
import scipy.fftpack as F
import scipy.io as io

import cv2
import matplotlib.image as img

from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from matplotlib import cm

import skimage
import skimage.morphology
import skimage.filters

from PIL import Image

import scipy.io as io
```

[2]:
```
# Importamos todas nuestras funciones:
import mfilt_funcs as mine
```

```
importlib.reload(mine)
from mfilt_funcs import *
```

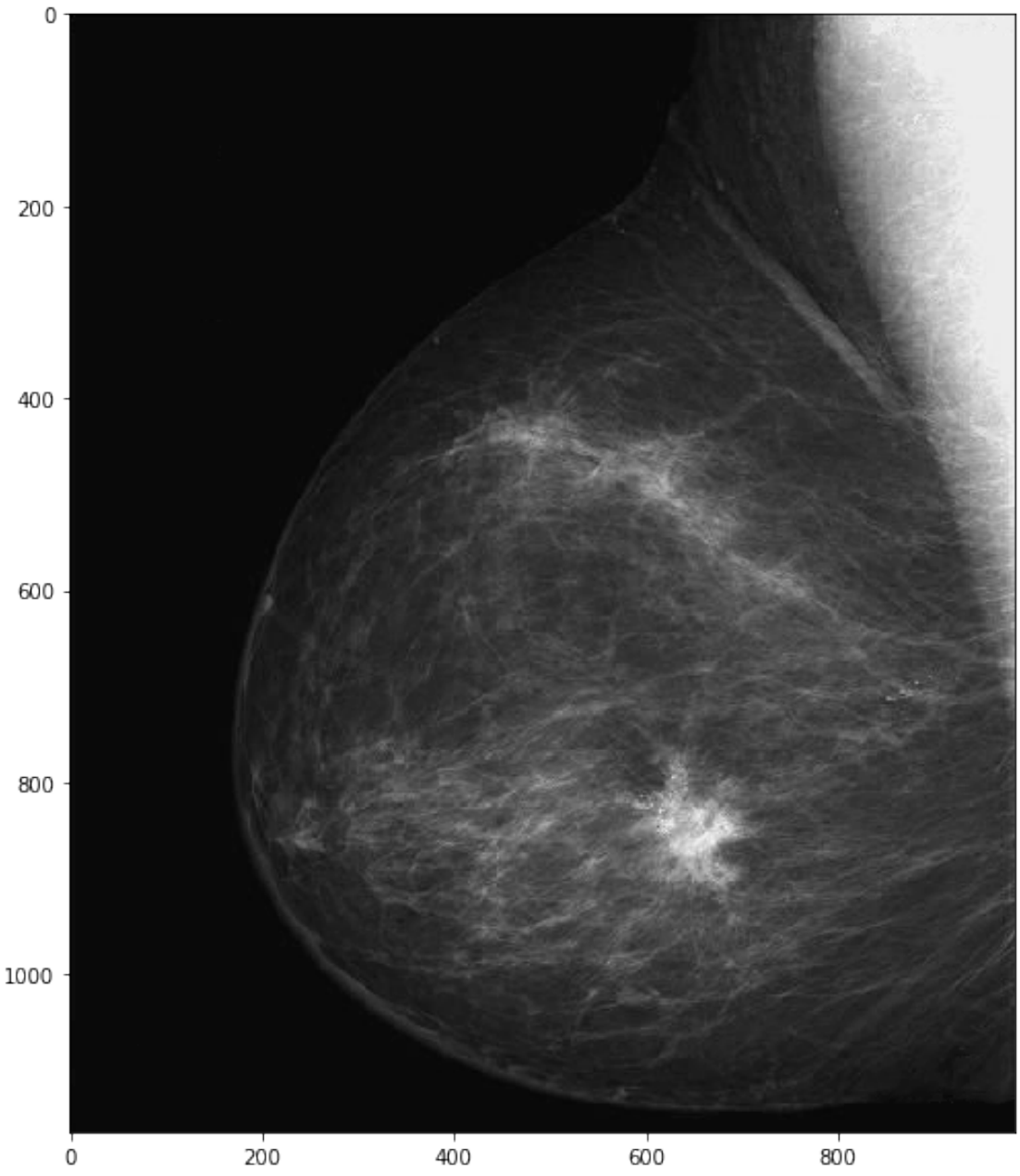
[3]:
```
def plot_all(image, **kw):
    """
    """

    fig = plt.figure(figsize = (15, 10))
    H = master_kernel(image, **kw)
    filtrada = filtra_maestra(image, **kw)

    fig.add_subplot(2, 3, 1)
    plt.imshow(image, cmap = 'gray')
    plt.title('Imagen original', size = 18)

    fig.add_subplot(2, 3, 2)
    fft_viz(image)
    plt.title('Transformada de la imagen', size = 18)

    fig.add_subplot(2, 3, 3)
    plt.imshow(filtrada, cmap = 'gray')
    plt.title(f'Imagen filtrada.', size = 18)
```

[4]:
```
plt.rcParams['figure.figsize'] = (10, 10)
```

[5]:
```
eps = np.finfo(float).eps
eps.setflags(write=False)
```

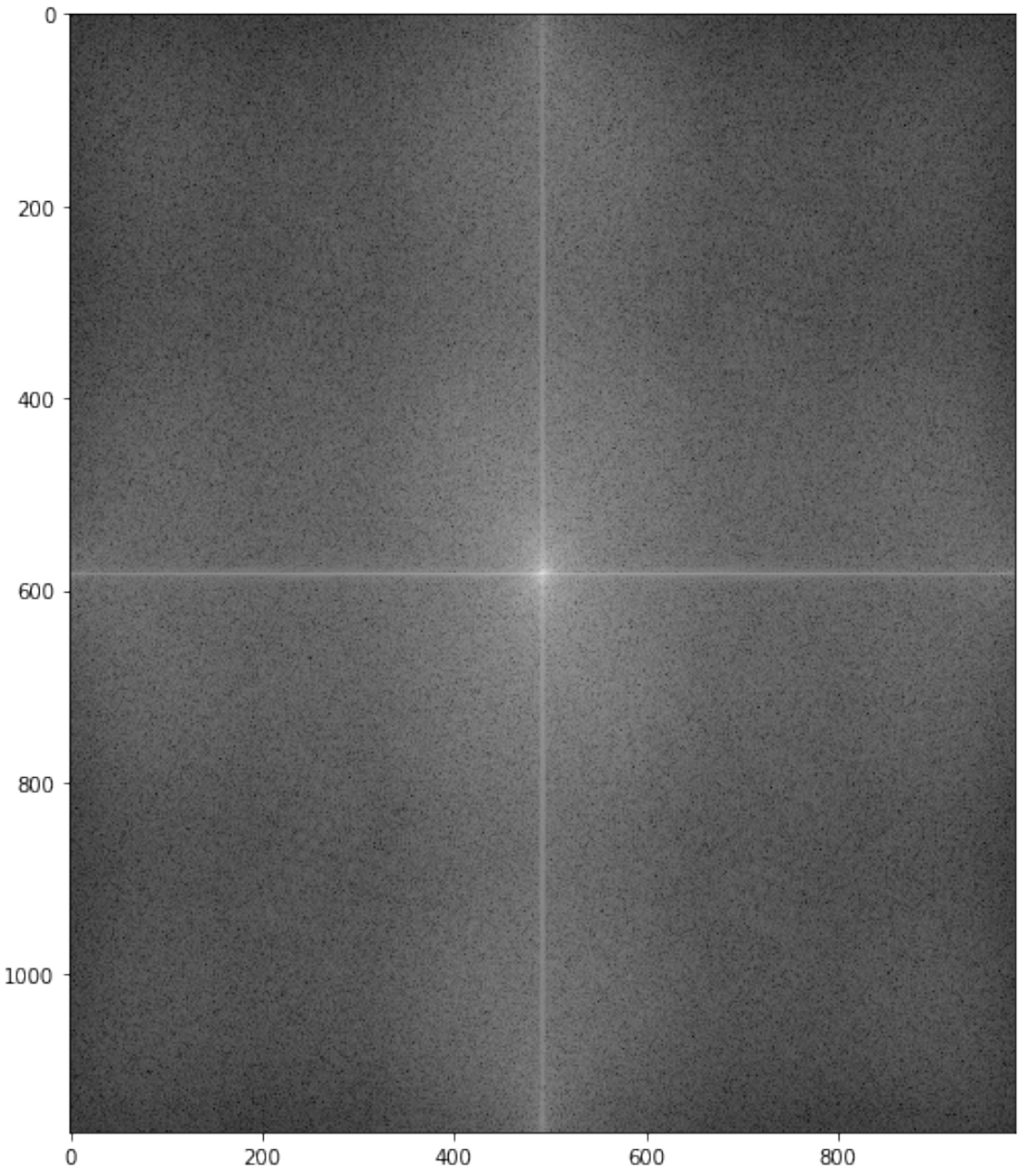
[6]:
```
I = img.imread('imagenes/mama.tif')
plt.imshow(I, cmap='gray')
```

[6]:
```
<matplotlib.image.AxesImage at 0x10d82f250>
```



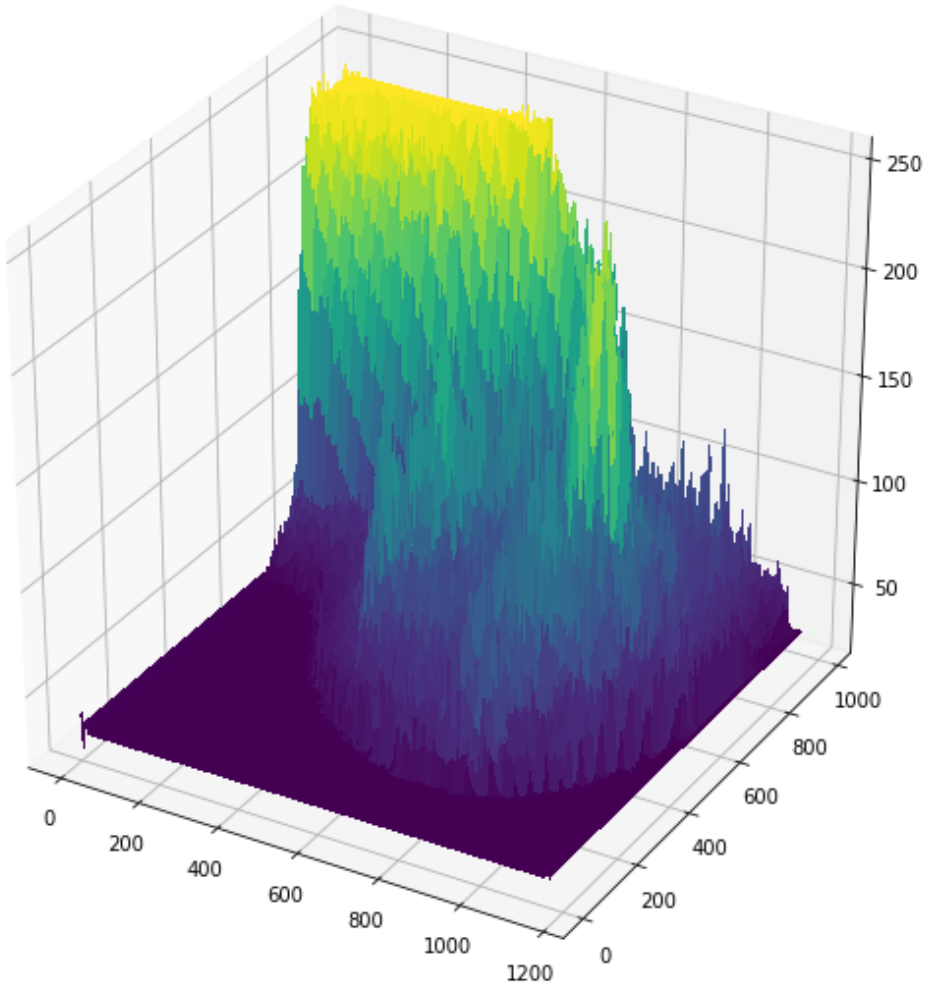
[7]:
```
fft_viz(I)
```

[7]:
```
<matplotlib.image.AxesImage at 0x1c2847a910>
```



[8]:
```
img_surf(I)
```

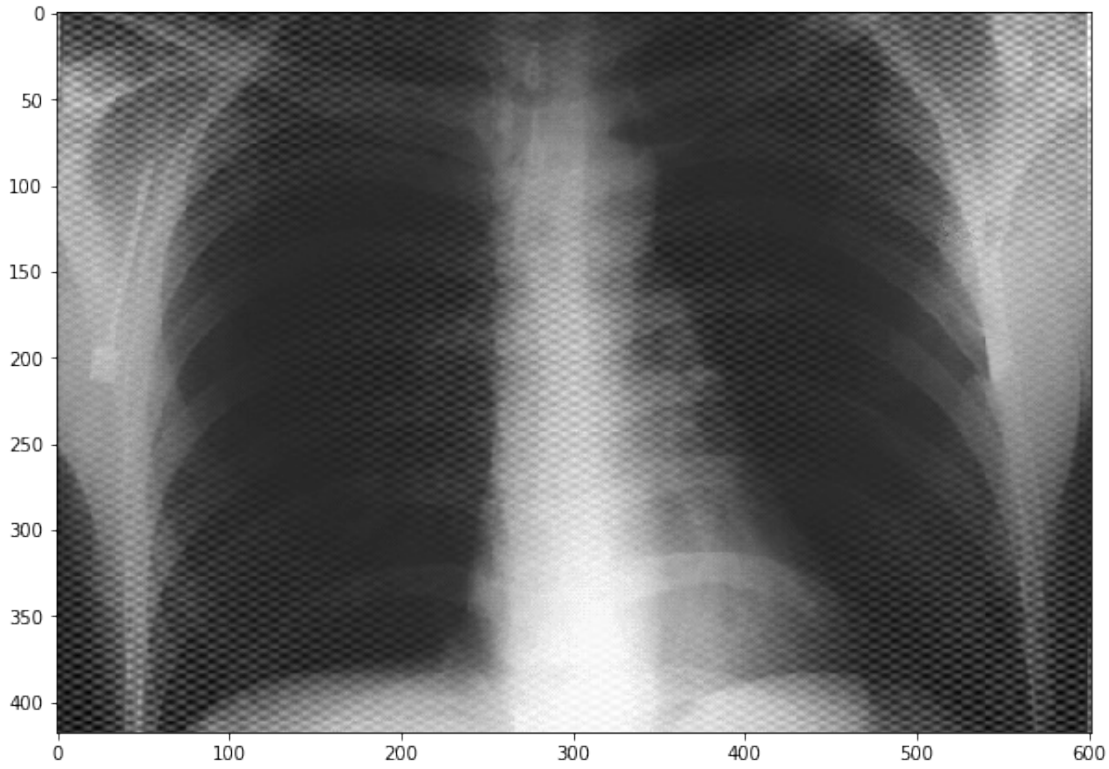
[8]:
```
<mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x1c29181c10>
```



[9]:
```
x = cv2.imread('imagenes/RadiografiaRuidoCoherente.jpg', 0)
```

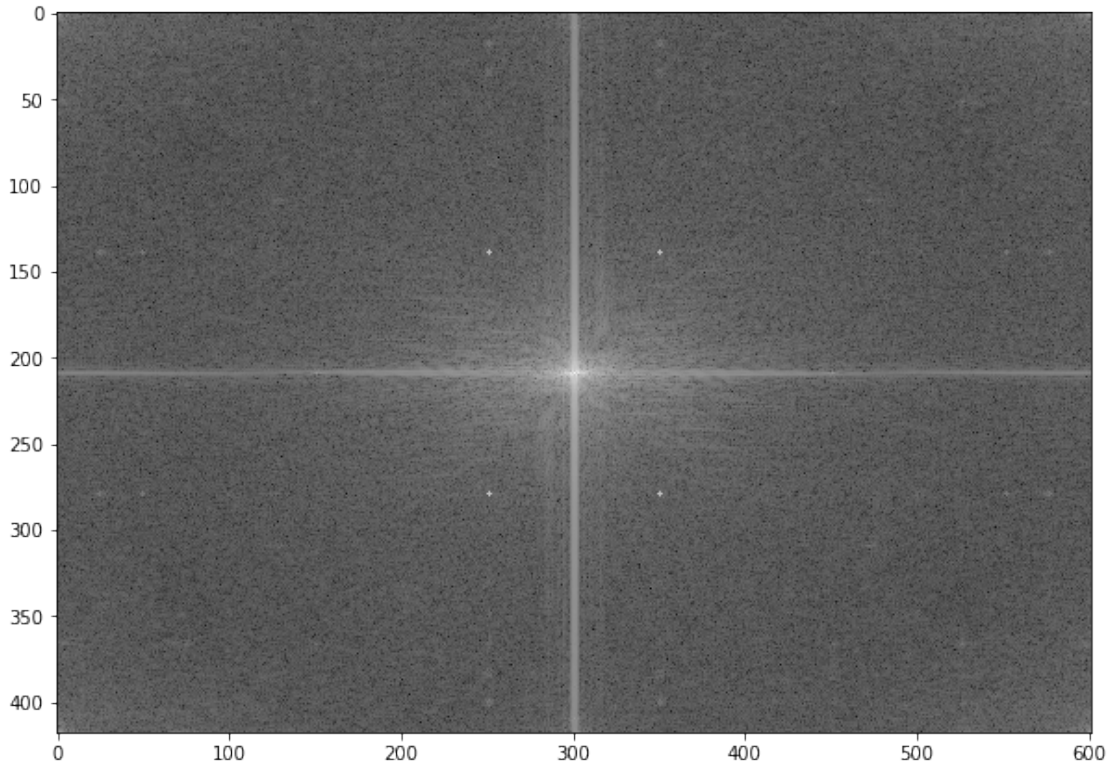
[10]:
```
plt.imshow(x, cmap='gray')
```

[10]:
```
<matplotlib.image.AxesImage at 0x1c2ac95490>
```



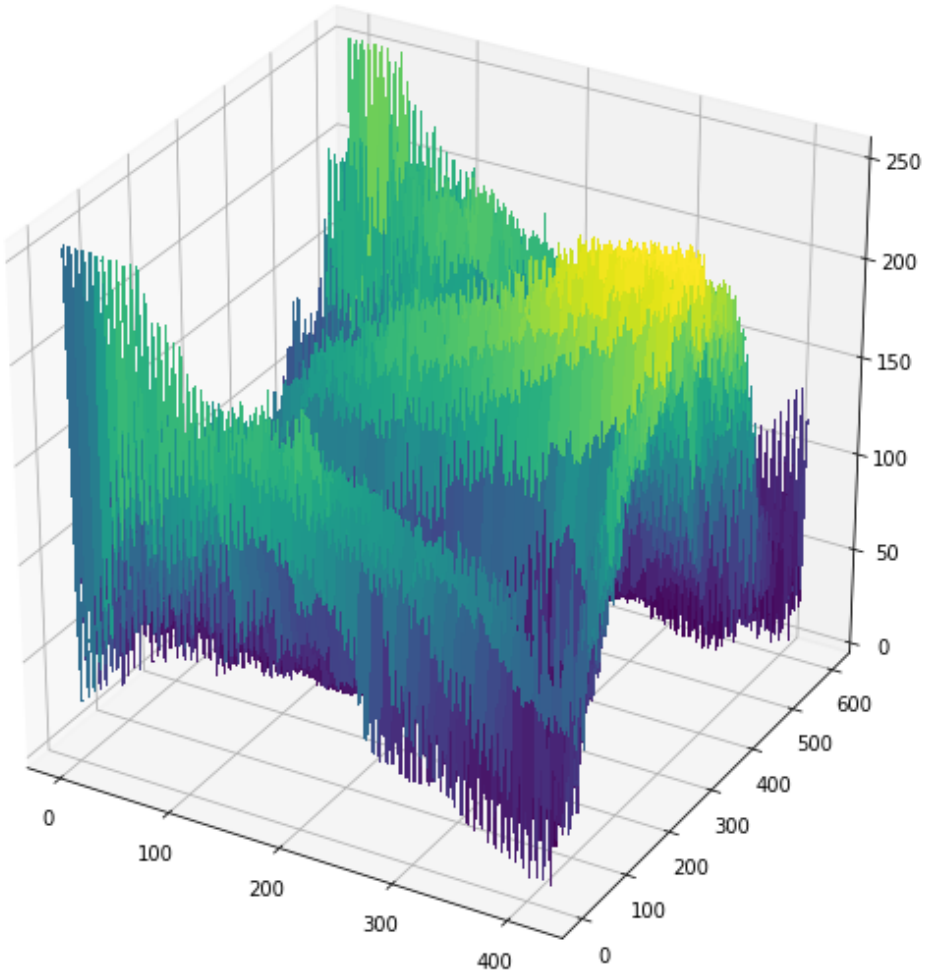
[11]:
```
fft_viz(x)
```

[11]: <matplotlib.image.AxesImage at 0x1c2af53d50>



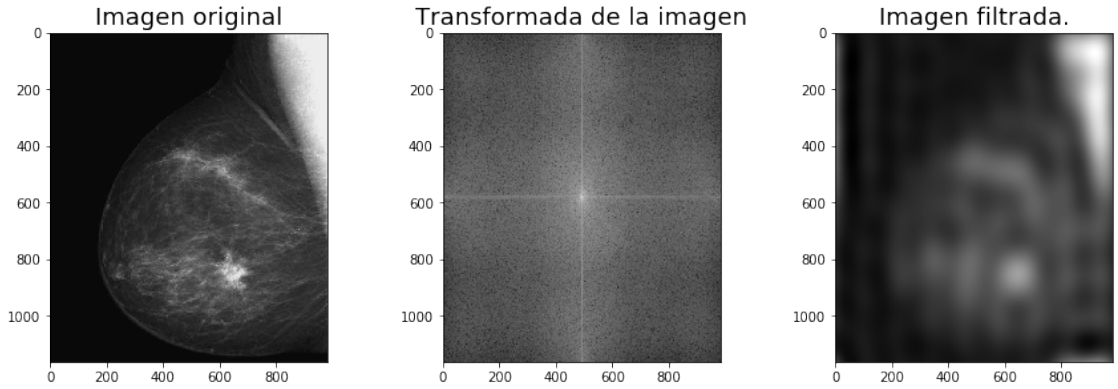
[12]:
```
img_surf(x)
```

[12]: <mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x1c2b0ec8d0>



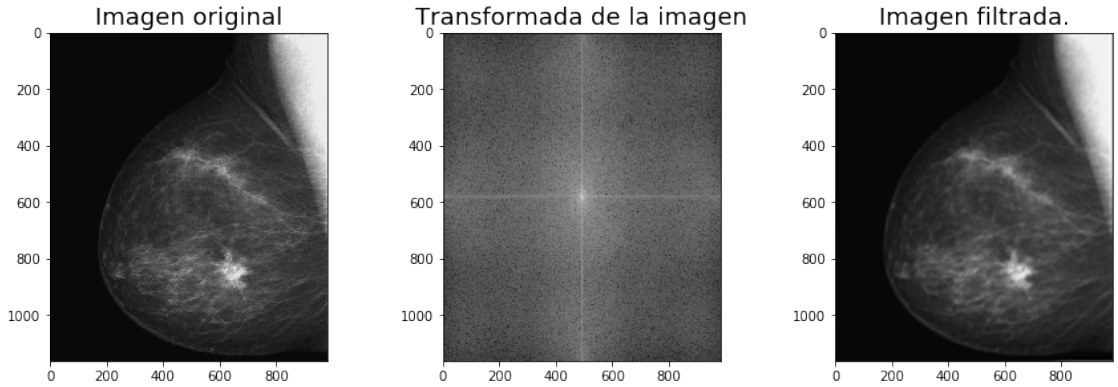
1.2.3 5.1 Filtro pasa bajos ideal con wc=64,

[13]:
```
banderas = dict(Do=64, kind='lowpass', form='ideal')
plot_all(I, **banderas)
```



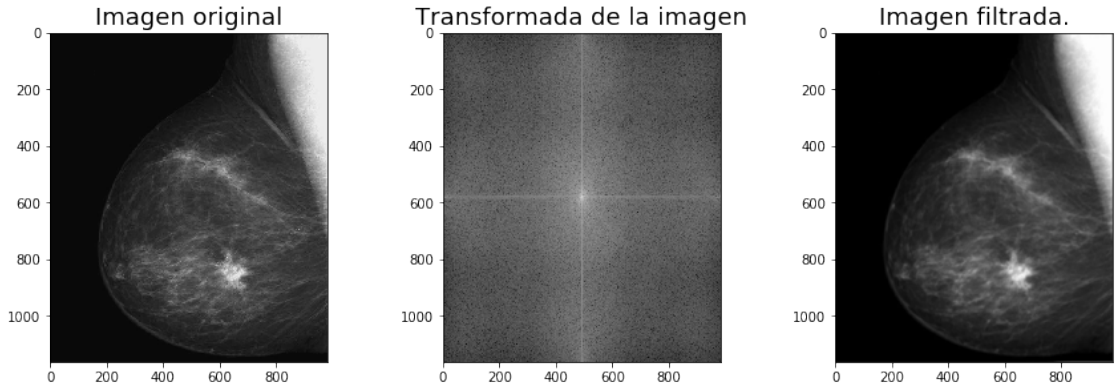
1.2.4 5.2 Filtro pasa bajos butt con wc=64, orden=2

[14]:
```
banderas = dict(Do=64, kind='lowpass', form='btw', n=2)
plot_all(I, **banderas)
```



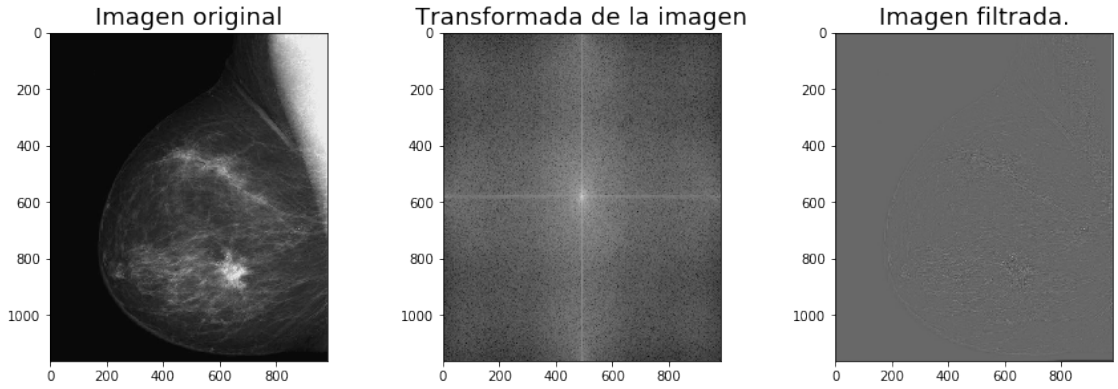
1.2.5 5.3 Filtro pasa bajos gauss con wc=64

[15]:
```
banderas = dict(Do=64, kind='lowpass', form='gauss')
plot_all(I, **banderas)
```



1.2.6 5.4 Filtro pasa altos gauss con wc=64

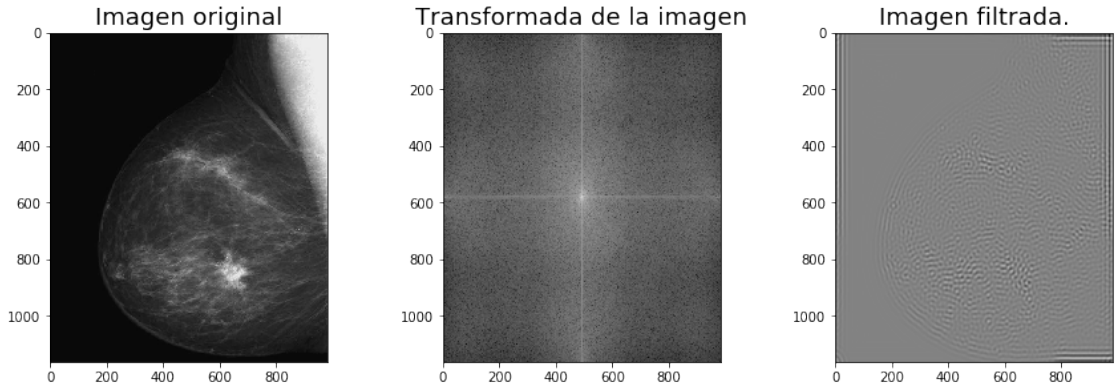
[16]:
```
banderas = dict(Do=64, kind='highpass', form='gauss')
plot_all(I, **banderas)
```



1.2.7 5.5 Filtro pasa bandas gauss con wc1=54, wc2=74

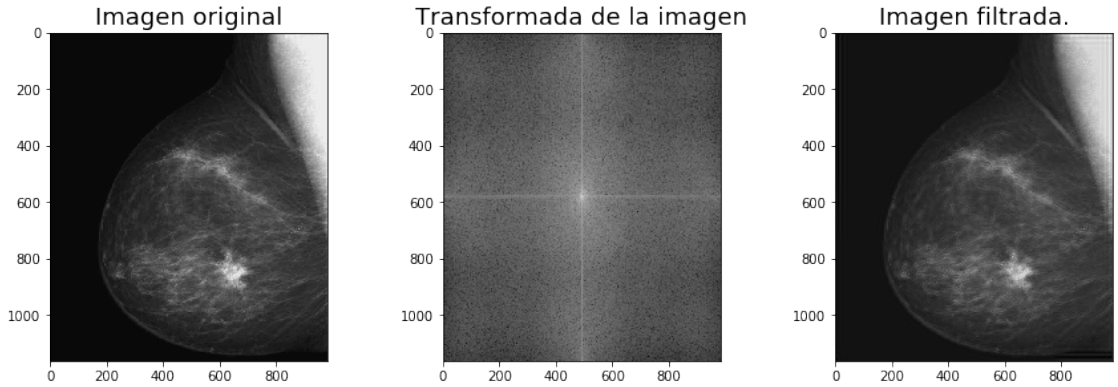
[17]:
```
banderas = dict(wc1=54, wc2=74, kind='bandpass', form='gauss')
plot_all(I, **banderas)
```

```
/Users/gml/Documents/IX/imagenes/MasterFilter/mfilt_funcs.py:400:
RuntimeWarning: divide by zero encountered in true_divide
  H = 1.0 - np.exp(-1.0 * (D - Do**2)**2 / (w**2 * D) )
```



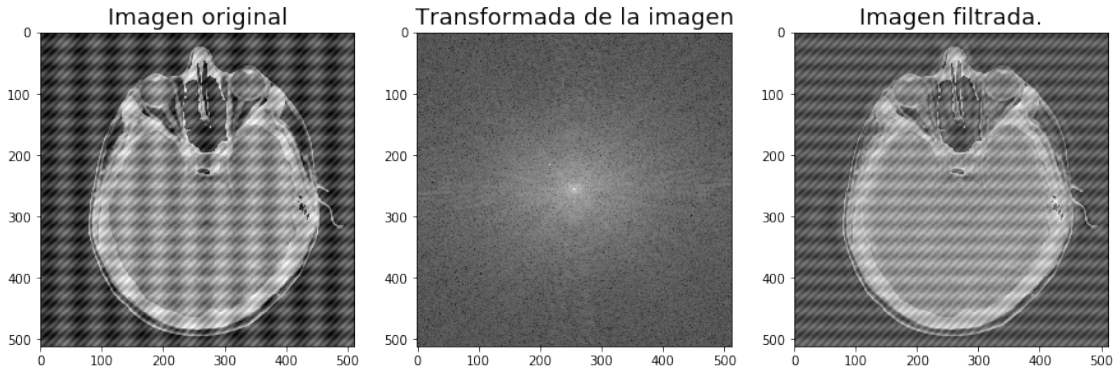
1.2.8 5.6 Filtro rechazo de bandas gauss con wc1=54, wc2=74

[18]:
```
banderas = dict(wc1=54, wc2=74, kind='bandreject', form='gauss')
plot_all(I, **banderas)
```



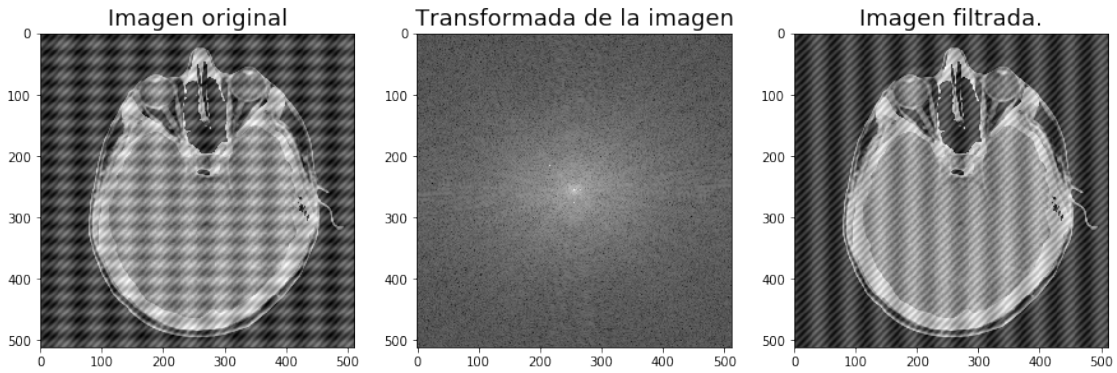
[19]:
```
cabeza = plt.imread('imagenes/FigP0405(HeadCT_corrupted).tif')
```

[20]:
```
banderas = dict(kind='notchreject', Do=15, center=(0, 25))
plot_all(cabeza, **banderas)
```



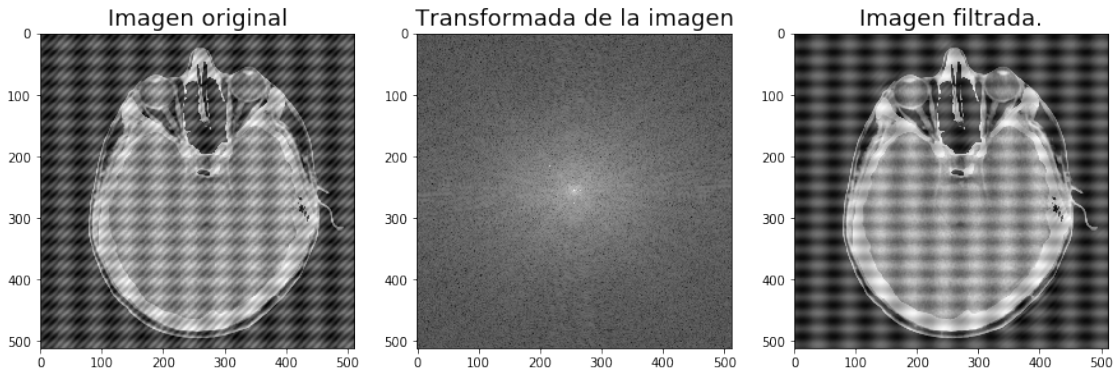
Eliminamos ruido vertical

[22]:
```
banderas = dict(kind='notchreject', Do=5, center=(20, 0), form='gauss')
plot_all(cabeza, **banderas)
```



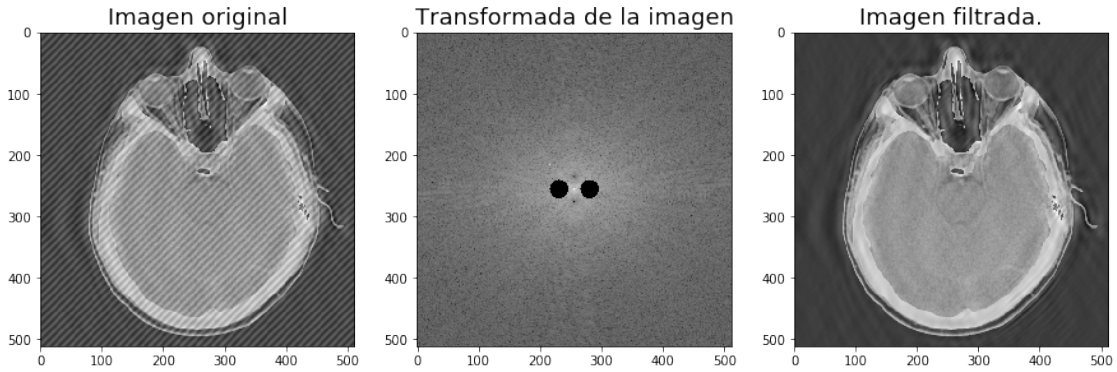
Eliminamos ruido horizontal.

[24]:
```
banderas = dict(kind='notchreject', Do=15, center=(46, 46))
plot_all(cabeza, **banderas)
```



Eliminamos ruido diagonal.

[25]:
```
banderas_vertical   = dict(kind='notchreject', Do=15, center=(0, 25))
banderas_horizontal = dict(kind='notchreject', Do=5, center=(20, 0),␣
 ↪form='gauss')
banderas_diagonal   = dict(kind='notchreject', Do=15, center=(46, 46))
pre = filtra_maestra(
    filtra_maestra(
        cabeza, **banderas_vertical
    ), **banderas_horizontal
)
plot_all(pre, **banderas_diagonal)
```



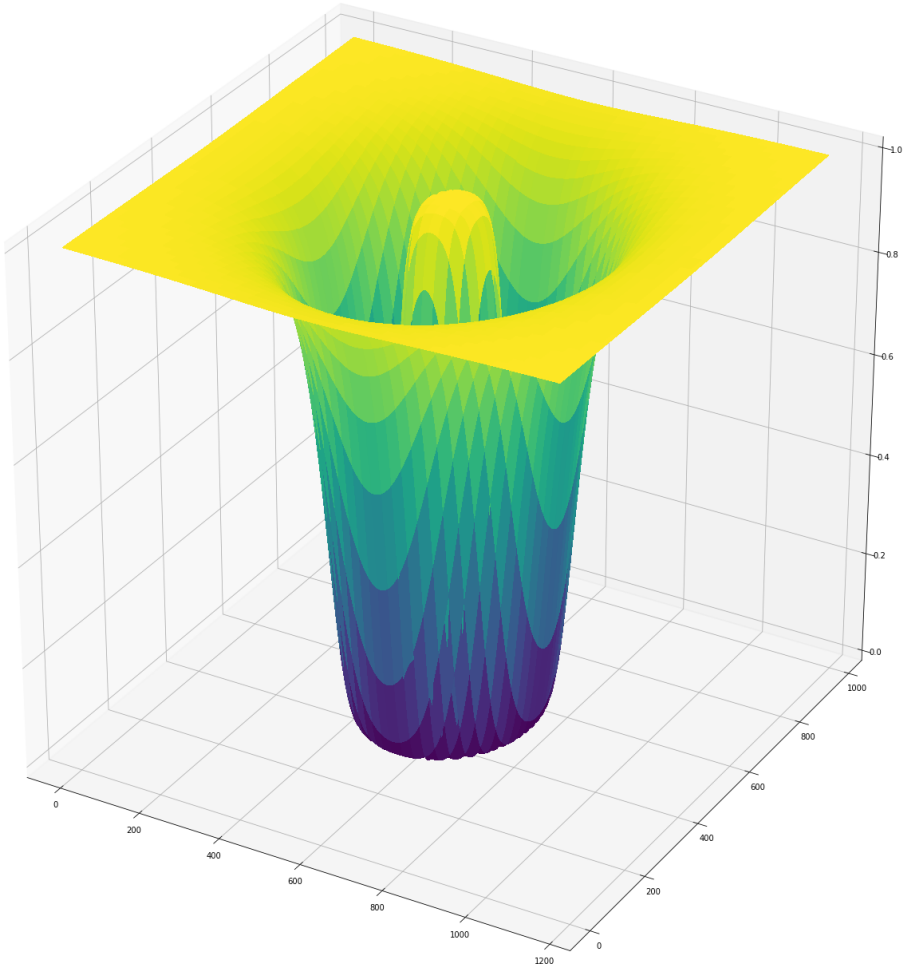
Con un poco de deformaión se logró eliminar el ruido periódico de la imagen.

1.2.9 Ejemplo de img_surf

Función definida en mfilt_funcs.py, permite representar una imagen (numpy.ndarray) como una superficie.

[73]:
```
img_surf(master_kernel(I, kind='bandreject', form='btw', wc1=100, wc2=250, n=2))
```

[73]:
```
<mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x1c22b22d50>
```


1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
1200
0
200
400
600
800
1000

[ ]:

[ ]:

[ ]: