

```
In [77]: %matplotlib inline
import os
import sys
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from PIL import Image
import timeit
import cv2
import scipy
from scipy.ndimage import convolve
from scipy.ndimage import gaussian_filter
import imageio
```

```
In [34]: def read_image_convert_to_grayscale(IMG_NAME):
    # YOUR CODE HERE
    img = cv2.imread(IMG_NAME)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    gray = gray.astype(np.float64)

    return gray
```

```
In [35]: def read_image(IMG_NAME):
    img = cv2.imread(IMG_NAME)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    return img
```

```
In [73]: def get_low_res_img(IMG):
    img = cv2.resize(IMG, (0,0), fx=0.2, fy=0.2)

    return img
```

```
In [36]: def filter_images(IMG,filter_type,sigma):

    if (filter_type == 'low_pass'):
        filter_img = gaussian_filter(IMG,sigma)
    elif (filter_type == 'high_pass'):
        filter_img = IMG - gaussian_filter(IMG,sigma)
    else:
        print('not supported')
    return filter_img
```

```
In [37]: def create_hybrid_images(img_smooth,img_texture,hybrid_type):  
  
    if (hybrid_type == 'add'):  
        hybrid_imag = img_smooth + img_texture  
    elif(hybrid_type == 'average'):  
        hybrid_imag = 0.5*(img_smooth + img_texture)  
    else:  
        print("not supproted")  
  
    return hybrid_imag
```

```
In [83]: IMG_DIR = 'part1_images/'  
IMG_NAME_smooth = 'c1.jpg'  
IMG_NAME_text = 'c2.jpg'  
sigma_low = 2  
sigma_high = 10  
  
smooth_imag = read_image_convert_to_grayscale(IMG_DIR + IMG_NAME_smooth)  
smooth_imag_org = read_image(IMG_DIR+IMG_NAME_smooth)  
texttured_imag = read_image_convert_to_grayscale(IMG_DIR + IMG_NAME_text)  
texttured_imag_org = read_image(IMG_DIR+IMG_NAME_text)  
  
filter_smooth_img = filter_images(smooth_imag,'low_pass',sigma_low)  
filter_textured_img = filter_images(texttured_imag,'high_pass',sigma_high)  
hybrid_img = create_hybrid_images(filter_smooth_img, filter_textured_img, 'add')  
  
hybrid_img_low_res = get_low_res_img(hybrid_img)  
  
output_folder = 'output/'  
plt.figure(1,figsize=[20 , 10])  
plt.imshow(smooth_imag_org)  
plt.show()  
cv2.imwrite(output_folder + 'smooth_org_' + IMG_NAME_smooth,cv2.cvtColor(smooth_imag_org, cv2.COLOR_RGB2BGR))  
  
plt.figure(2,figsize=[20 , 10])  
plt.imshow(texttured_imag_org)  
plt.show()  
cv2.imwrite(output_folder + 'text_org_' + IMG_NAME_text,cv2.cvtColor(texttured_imag_org, cv2.COLOR_RGB2BGR))
```

```
plt.figure(3,figsize=[20 , 10])
plt.imshow(filter_smooth_img,cmap = 'gray')
cv2.imwrite(output_folder + 'smooth_filter_'+IMG_NAME_smooth,filter_smooth_img)

plt.figure(4,figsize=[20 , 10])
plt.imshow(filter_textured_img, cmap = 'gray')
plt.show()
#cv2.imwrite(output_folder + 'text_filter_'+IMG_NAME_text,filter_textured_img)
imageio.imwrite(output_folder + 'text_filter_'+IMG_NAME_text, filter_textured_img)

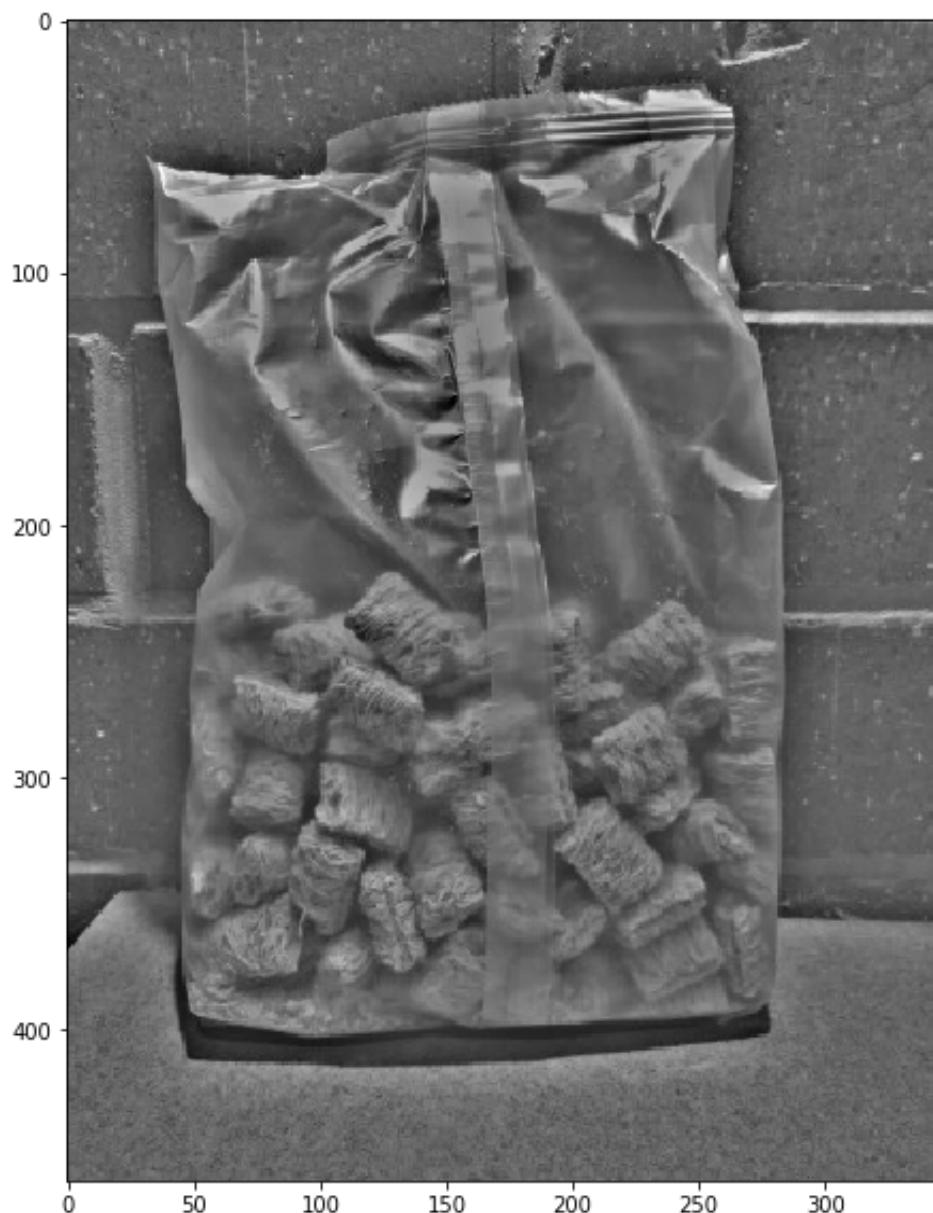
plt.figure(5,figsize=[20 , 10])
plt.imshow(hybrid_img, cmap = 'gray')
plt.show()
cv2.imwrite(output_folder + 'hybrd_hig_res_'+IMG_NAME_text,hybrid_img)

plt.figure(6,figsize=[20 , 10])
plt.imshow(hybrid_img_low_res, cmap = 'gray')
cv2.imwrite(output_folder + 'hybrd_low_res_'+IMG_NAME_text,hybrid_img_low_res)
```









WARNING:root:Lossy conversion from float64 to uint8. Range [-104.016
17442501953, 158.01340404107833]. Convert image to uint8 prior to sa
ving to suppress this warning.



Out[83]: True



```
In [86]: IMG_DIR = 'part1_images/'  
IMG_NAME_smooth = 'a1.jpg'  
IMG_NAME_text = 'a2.jpg'  
sigma_low = 2  
sigma_high = 10  
  
smooth_imag = read_image_convert_to_grayscale(IMG_DIR + IMG_NAME_smooth)  
smooth_imag_org = read_image(IMG_DIR+IMG_NAME_smooth)  
  
textttured_imag = read_image_convert_to_grayscale(IMG_DIR + IMG_NAME_text)  
textttured_imag_org = read_image(IMG_DIR+IMG_NAME_text)
```

```
filter_smooth_img = filter_images(smooth_imag, 'low_pass', sigma_low)
filter_textured_img = filter_images(textttured_imag, 'high_pass', sigma_high)
hybrid_img = create_hybrid_images(filter_smooth_img, filter_textured_img, 'add')

hybrid_img_low_res = get_low_res_img(hybrid_img)

output_folder = 'output/'
plt.figure(1, figsize=[20 , 10])
plt.imshow(smooth_imag_org)
plt.show()
cv2.imwrite(output_folder + 'smooth_org_' + IMG_NAME_smooth, cv2.cvtColor(smooth_imag_org, cv2.COLOR_RGB2BGR))

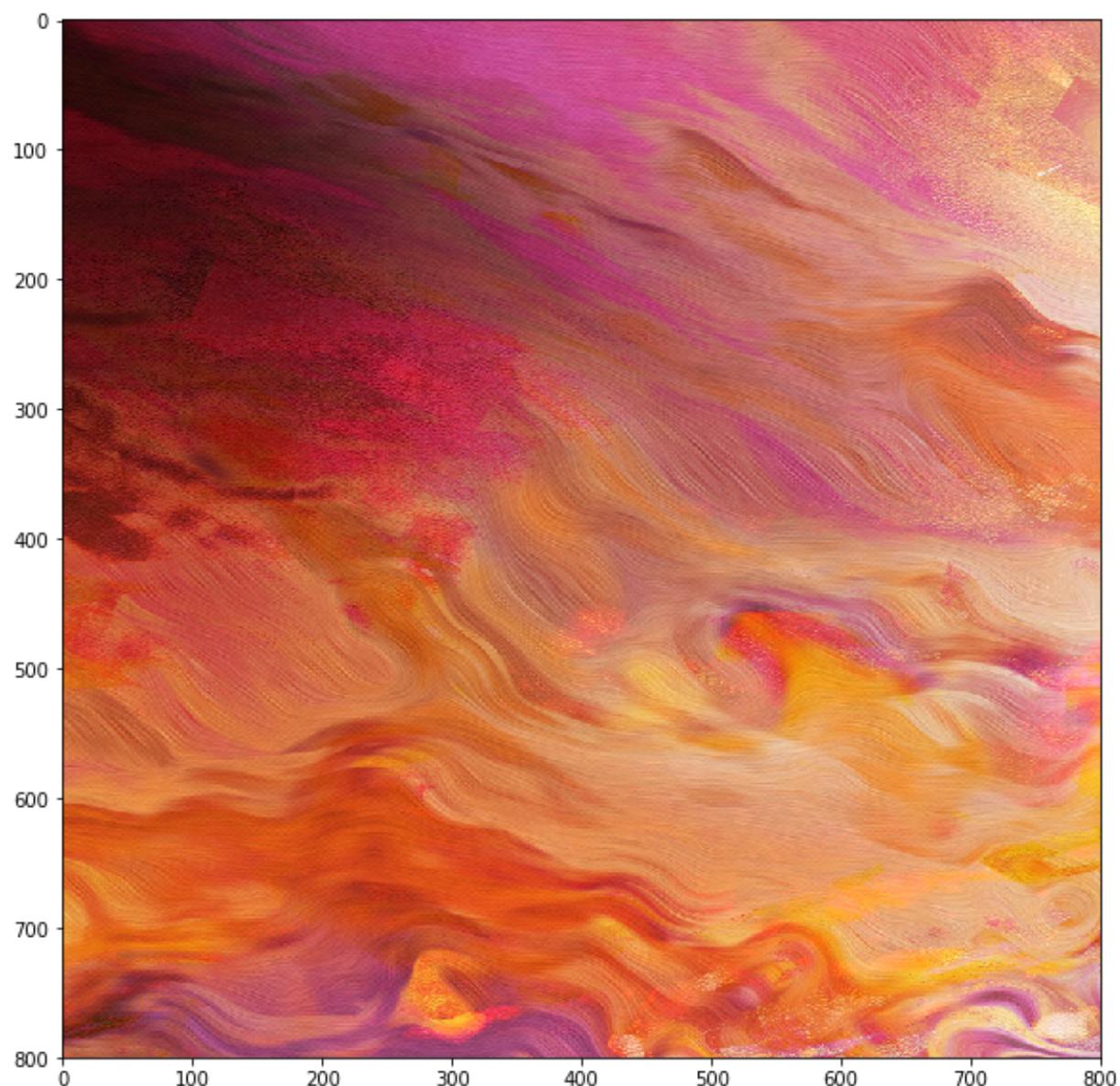
plt.figure(2, figsize=[20 , 10])
plt.imshow(textttured_imag_org)
plt.show()
cv2.imwrite(output_folder + 'text_org_' + IMG_NAME_text, cv2.cvtColor(texttured_imag_org, cv2.COLOR_RGB2BGR))

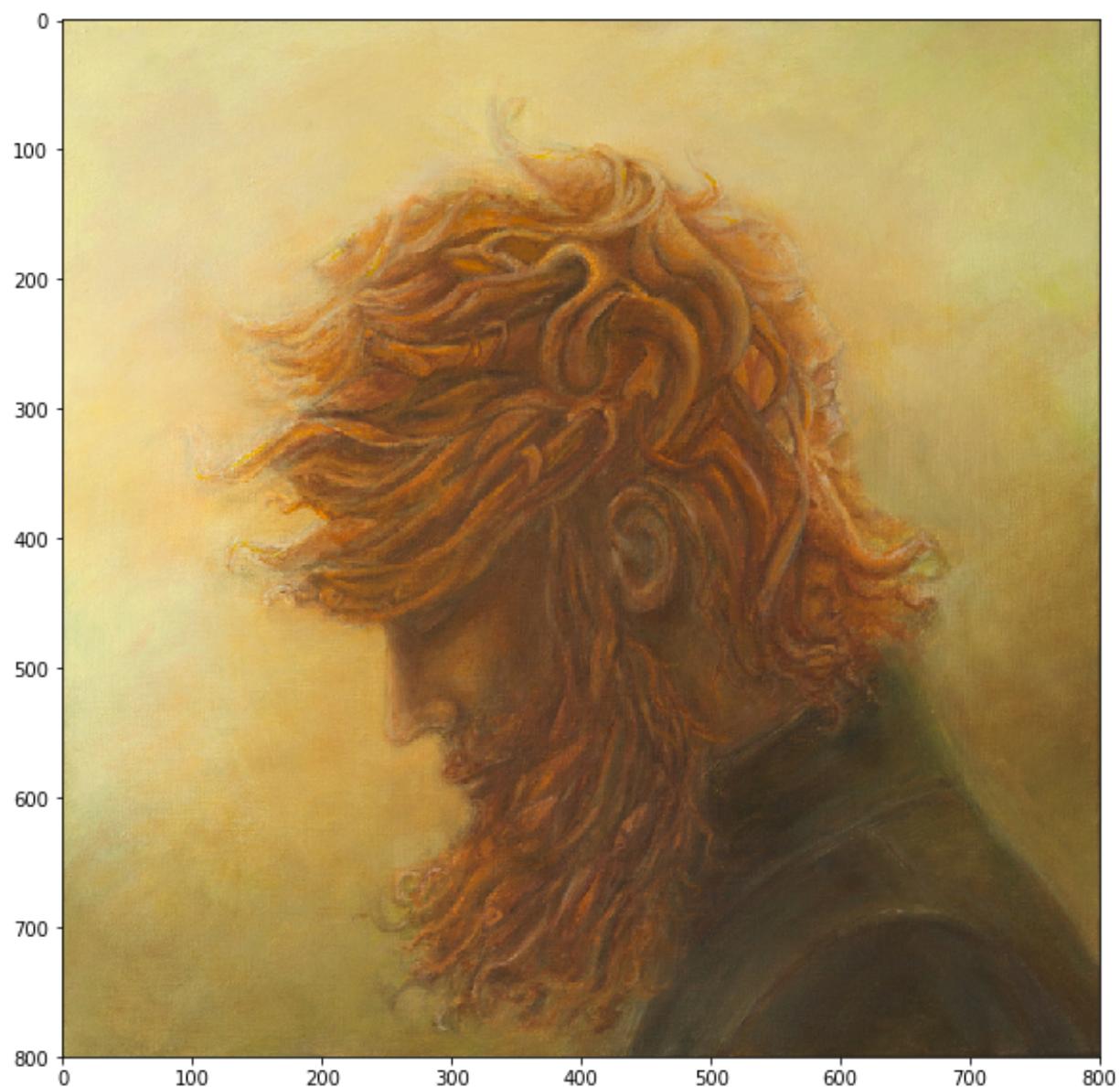
plt.figure(3, figsize=[20 , 10])
plt.imshow(filter_smooth_img, cmap = 'gray')
cv2.imwrite(output_folder + 'smooth_filter_'+IMG_NAME_smooth, filter_smooth_img)

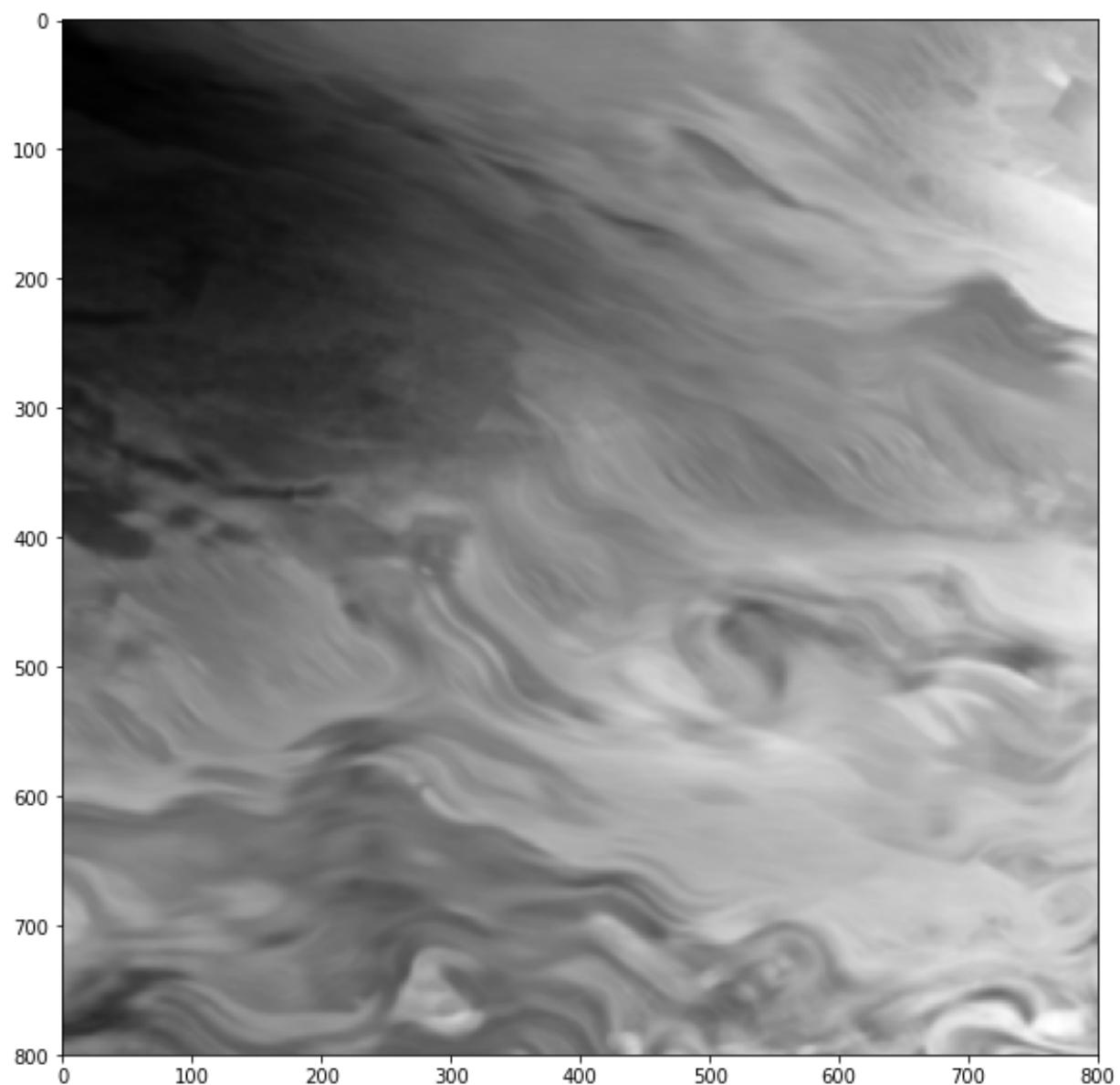
plt.figure(4, figsize=[20 , 10])
plt.imshow(filter_textured_img, cmap = 'gray')
plt.show()
#cv2.imwrite(output_folder + 'text_filter_'+IMG_NAME_text, filter_textured_img)
imageio.imwrite(output_folder + 'text_filter_'+IMG_NAME_text, filter_textured_img)

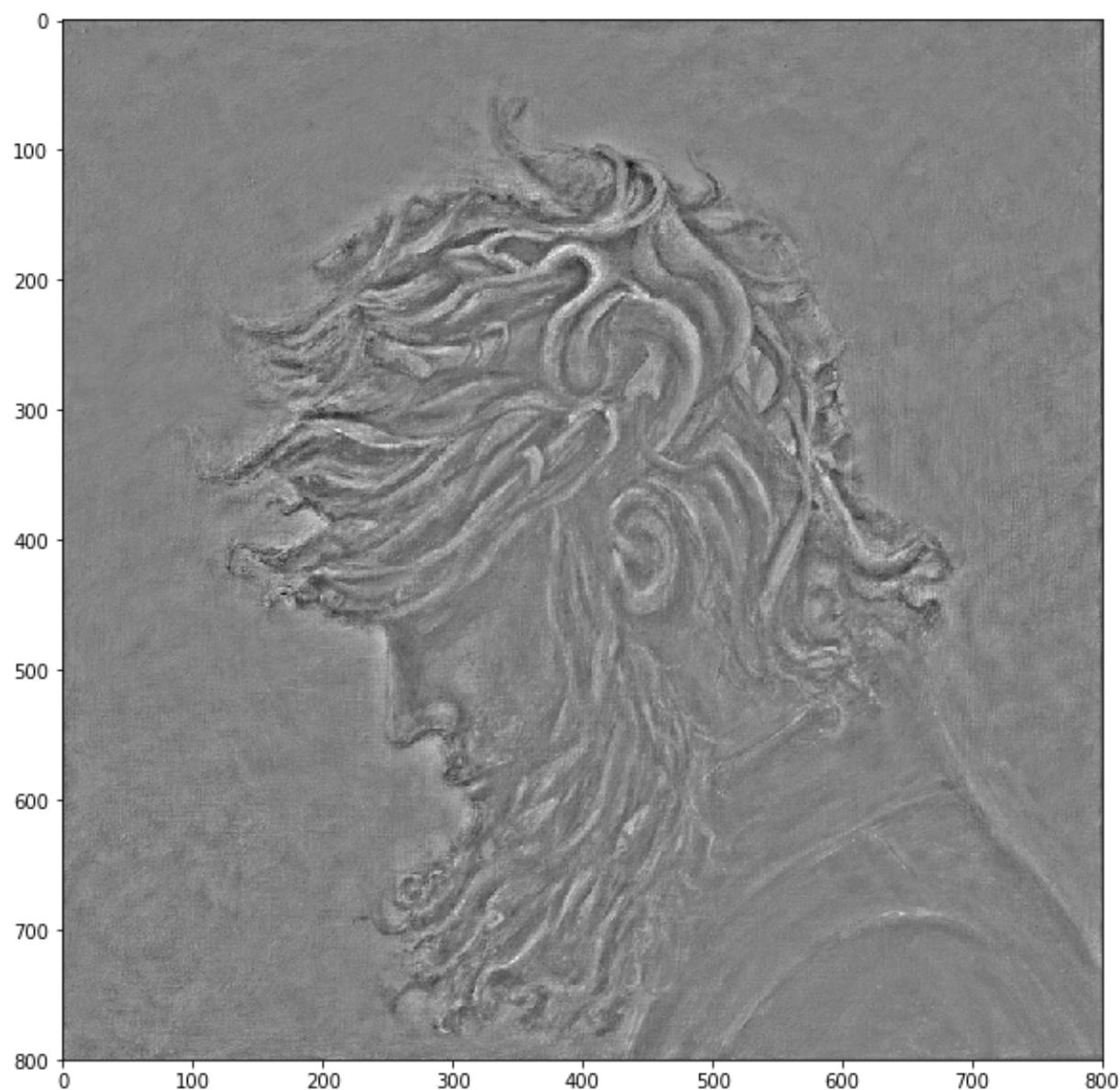
plt.figure(5, figsize=[20 , 10])
plt.imshow(hybrid_img, cmap = 'gray')
plt.show()
cv2.imwrite(output_folder + 'hybrd_hig_res_'+IMG_NAME_text, hybrid_img)

plt.figure(6, figsize=[20 , 10])
plt.imshow(hybrid_img_low_res, cmap = 'gray')
cv2.imwrite(output_folder + 'hybrd_low_res_'+IMG_NAME_text, hybrid_img_low_res)
```

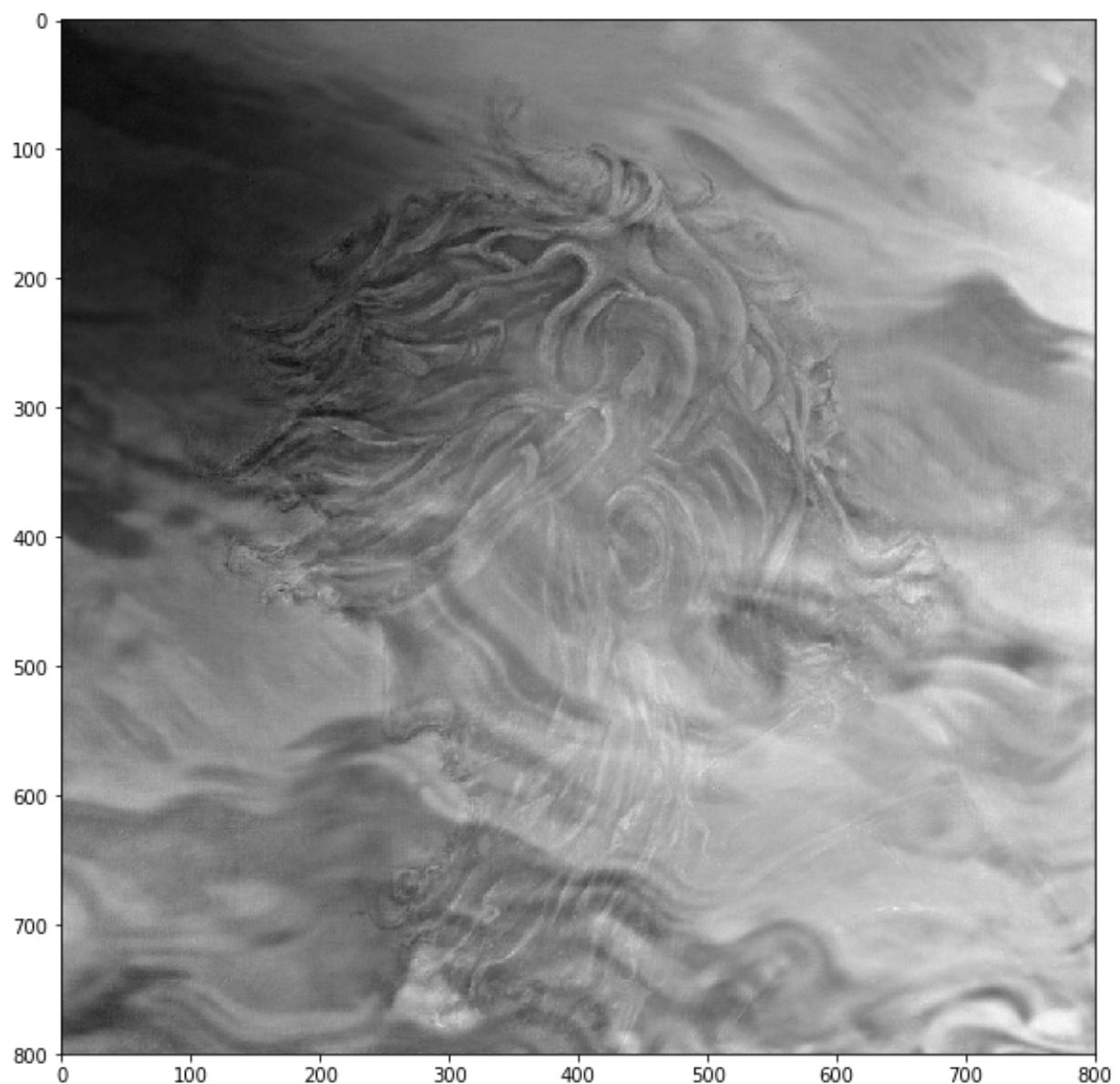




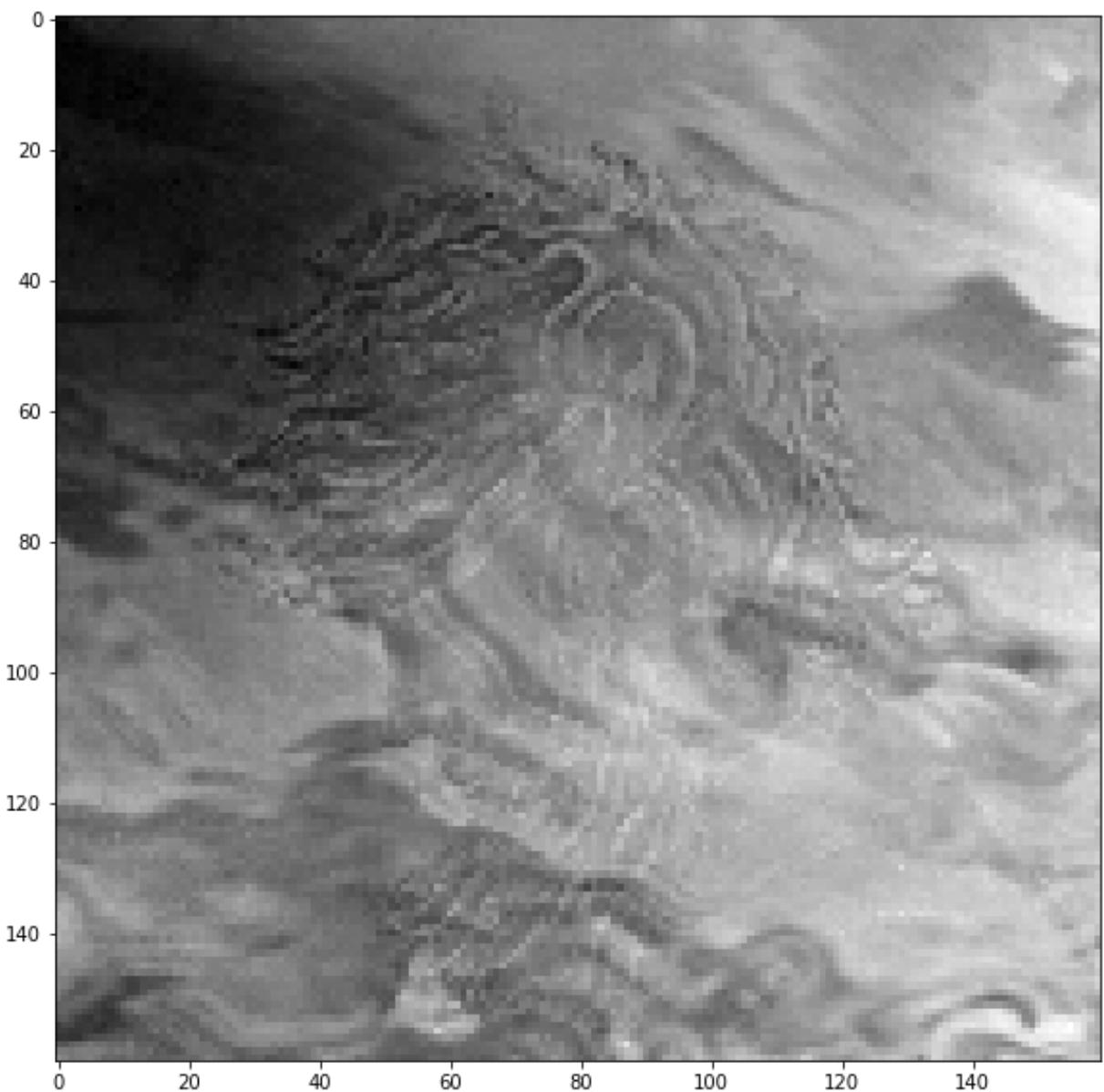




WARNING:root:Lossy conversion from float64 to uint8. Range [-63.5830 7915635592, 60.094082129872874]. Convert image to uint8 prior to saving to suppress this warning.



Out[86]: True



```
In [91]: IMG_DIR = 'part1_images/'  
IMG_NAME_smooth = 'b1.jpg'  
IMG_NAME_text = 'b2.jpg'  
sigma_low = 2  
sigma_high = 5  
  
smooth_imag = read_image_convert_to_grayscale(IMG_DIR + IMG_NAME_smooth)  
smooth_imag_org = read_image(IMG_DIR+IMG_NAME_smooth)  
  
texttured_imag = read_image_convert_to_grayscale(IMG_DIR + IMG_NAME_text)  
texttured_imag_org = read_image(IMG_DIR+IMG_NAME_text)
```

```
filter_smooth_img = filter_images(smooth_imag,'low_pass',sigma_low)
filter_textured_img = filter_images(textttured_imag,'high_pass',sigma_h
igh)
hybrid_img = create_hybrid_images(filter_smooth_img, filter_textured_i
mg, 'add')

hybrid_img_low_res = get_low_res_img(hybrid_img)

output_folder = 'output/'
plt.figure(1,figsize=[20 , 10])
plt.imshow(smooth_imag_org)
plt.show()
cv2.imwrite(output_folder + 'smooth_org_' + IMG_NAME_smooth,cv2.cvtColor(smooth_imag_org, cv2.COLOR_RGB2BGR))

plt.figure(2,figsize=[20 , 10])
plt.imshow(textttured_imag_org)
plt.show()
cv2.imwrite(output_folder + 'text_org_' + IMG_NAME_text,cv2.cvtColor(t
exttured_imag_org, cv2.COLOR_RGB2BGR))

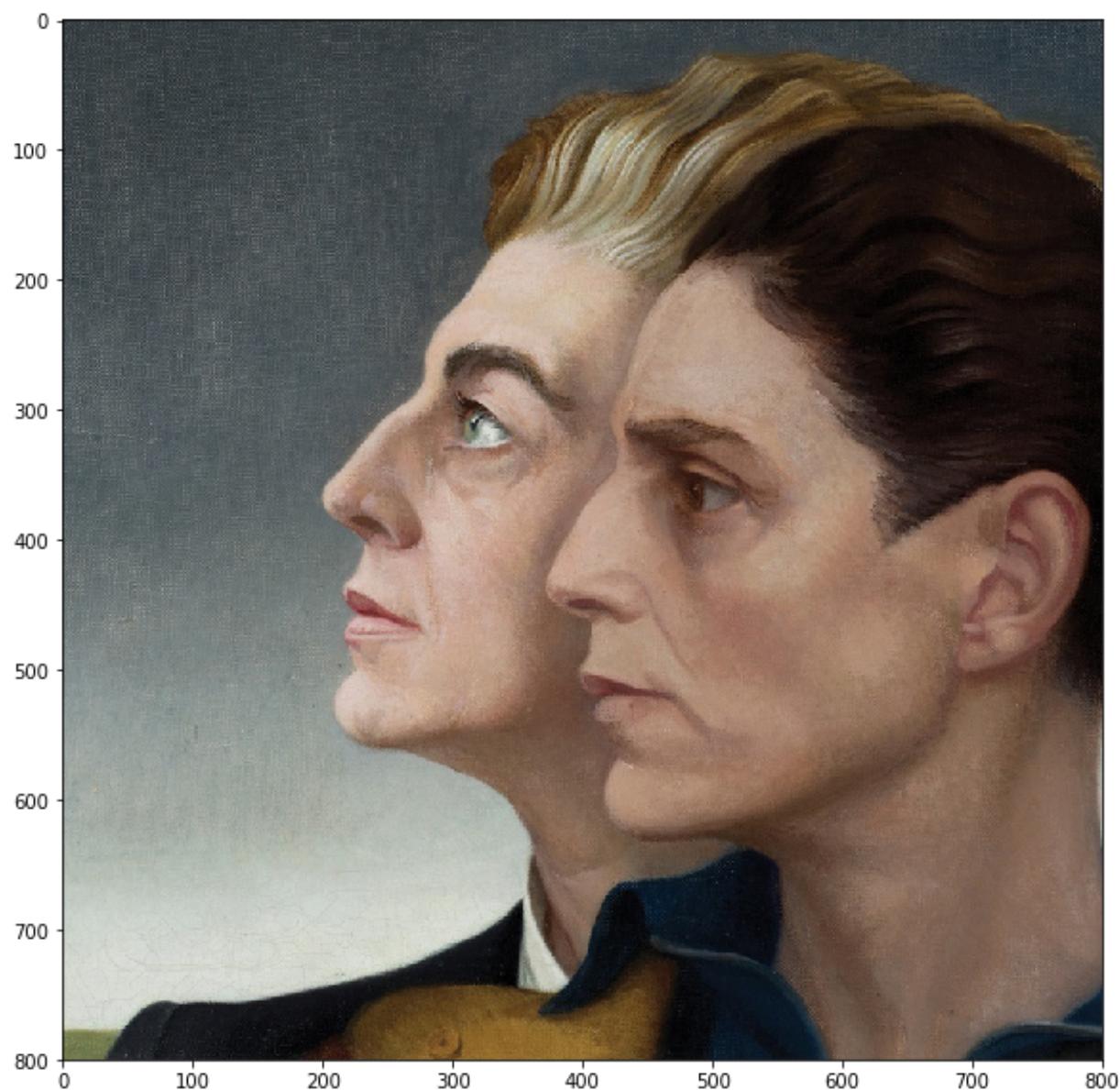
plt.figure(3,figsize=[20 , 10])
plt.imshow(filter_smooth_img,cmap = 'gray')
cv2.imwrite(output_folder + 'smooth_filter_'+IMG_NAME_smooth,filter_sm
ooth_img)

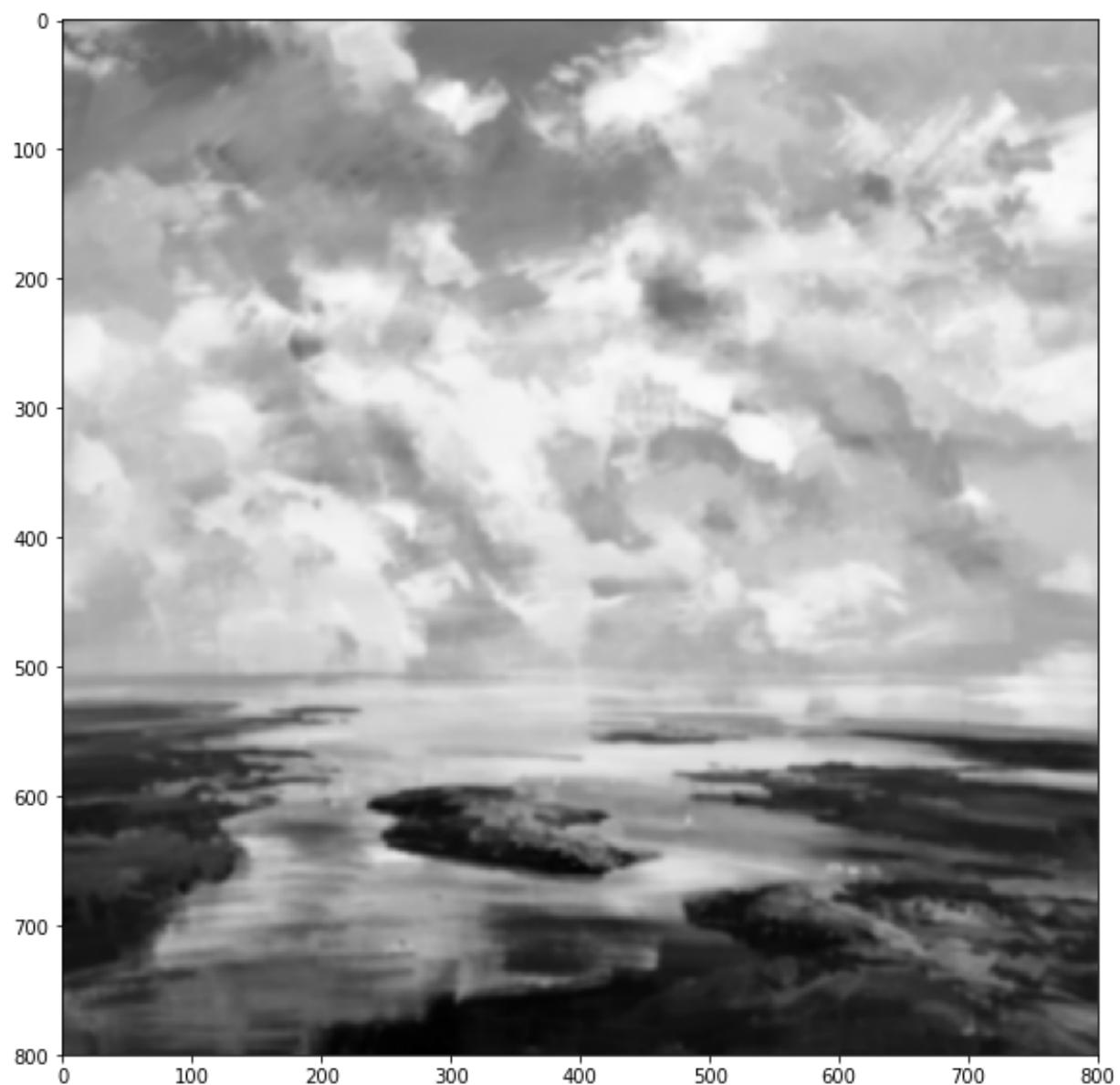
plt.figure(4,figsize=[20 , 10])
plt.imshow(filter_textured_img, cmap = 'gray')
plt.show()
#cv2.imwrite(output_folder + 'text_filter_'+IMG_NAME_text,filter_textu
red_img)
imageio.imwrite(output_folder + 'text_filter_'+IMG_NAME_text, filter_t
extured_img)

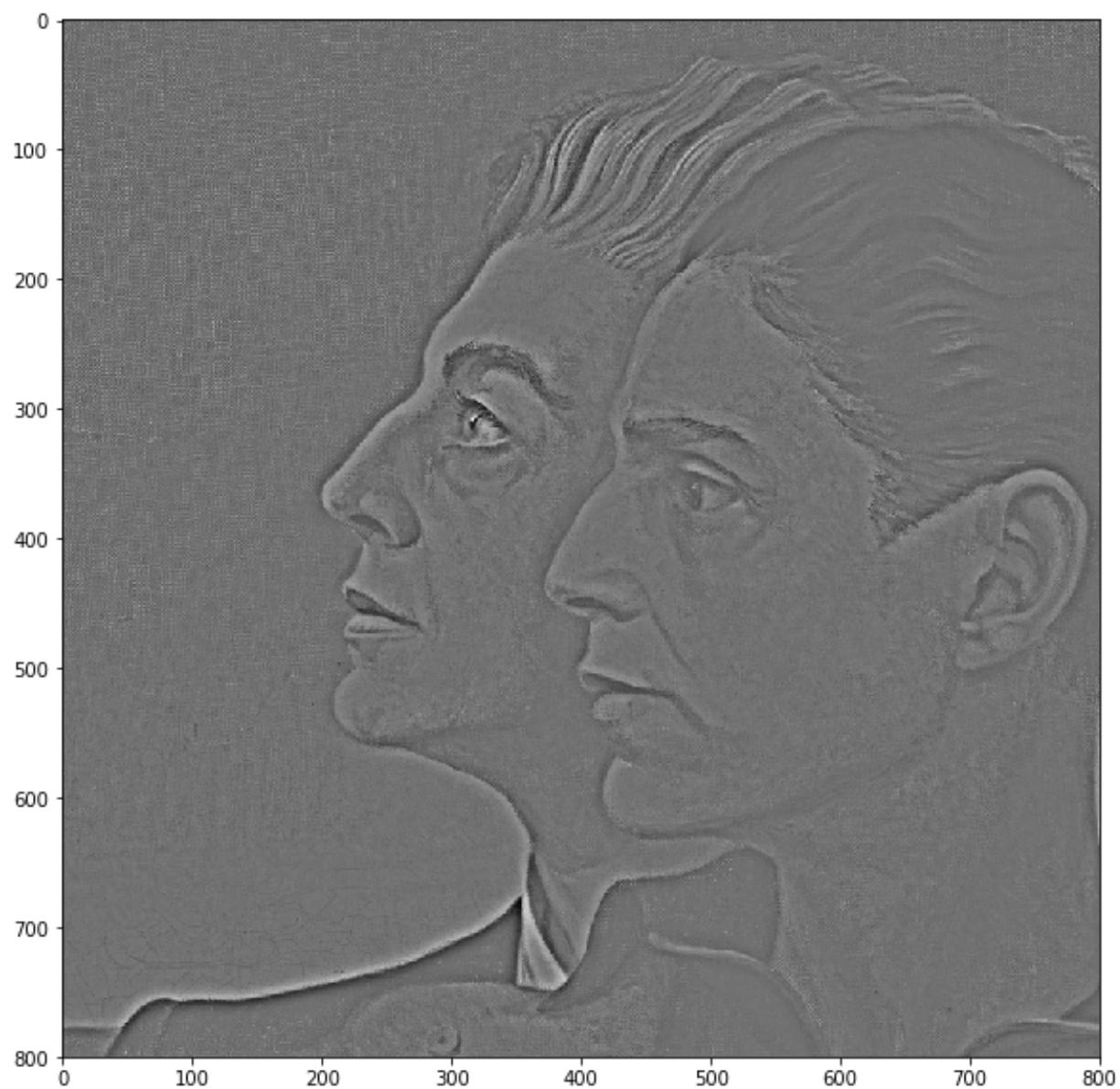
plt.figure(5,figsize=[20 , 10])
plt.imshow(hybrid_img, cmap = 'gray')
plt.show()
cv2.imwrite(output_folder + 'hybrd_hig_res_'+IMG_NAME_text,hybrid_img)

plt.figure(6,figsize=[20 , 10])
plt.imshow(hybrid_img_low_res, cmap = 'gray')
cv2.imwrite(output_folder + 'hybrd_low_res_'+IMG_NAME_text,hybrid_img_
low_res)
```

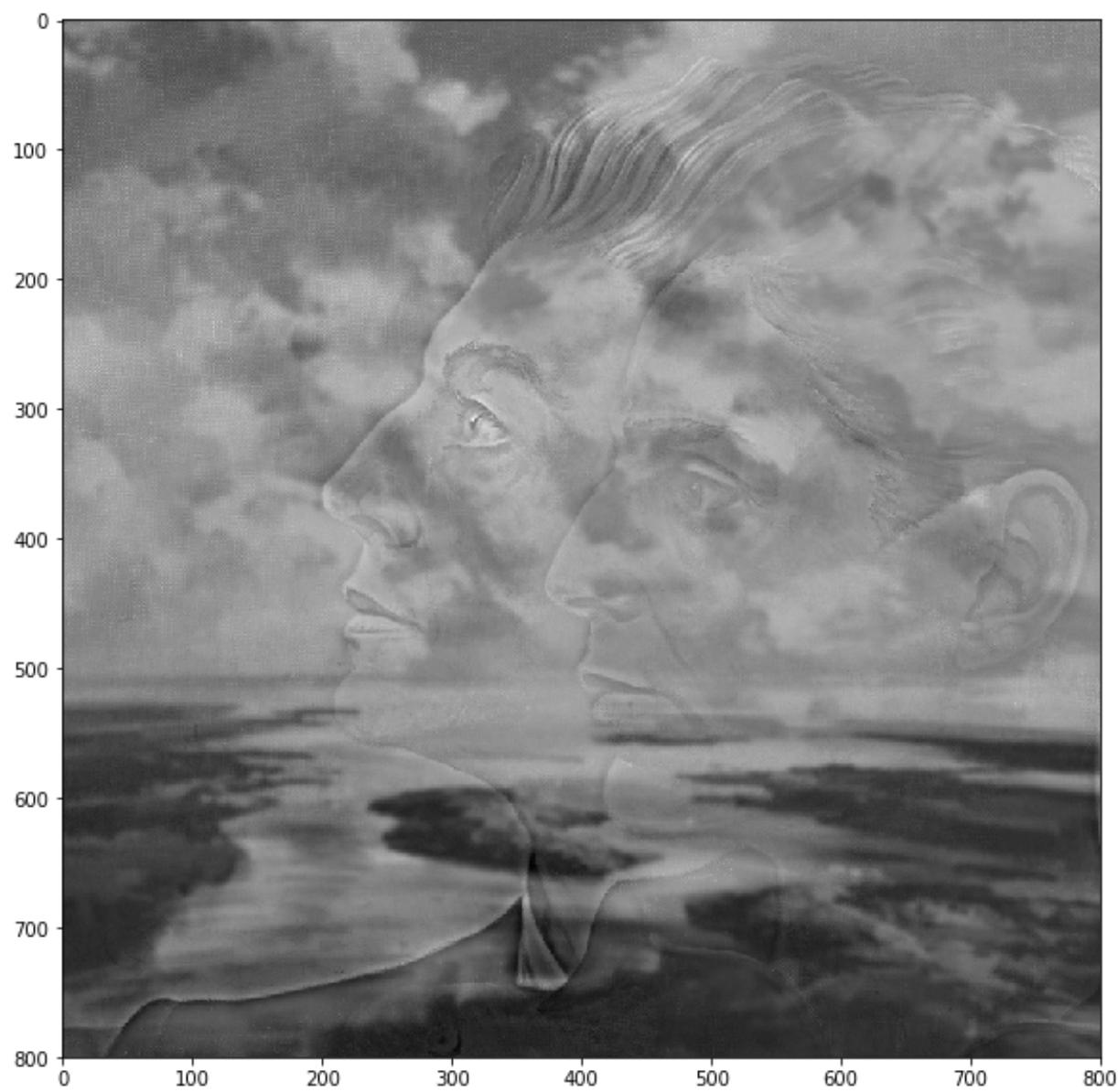




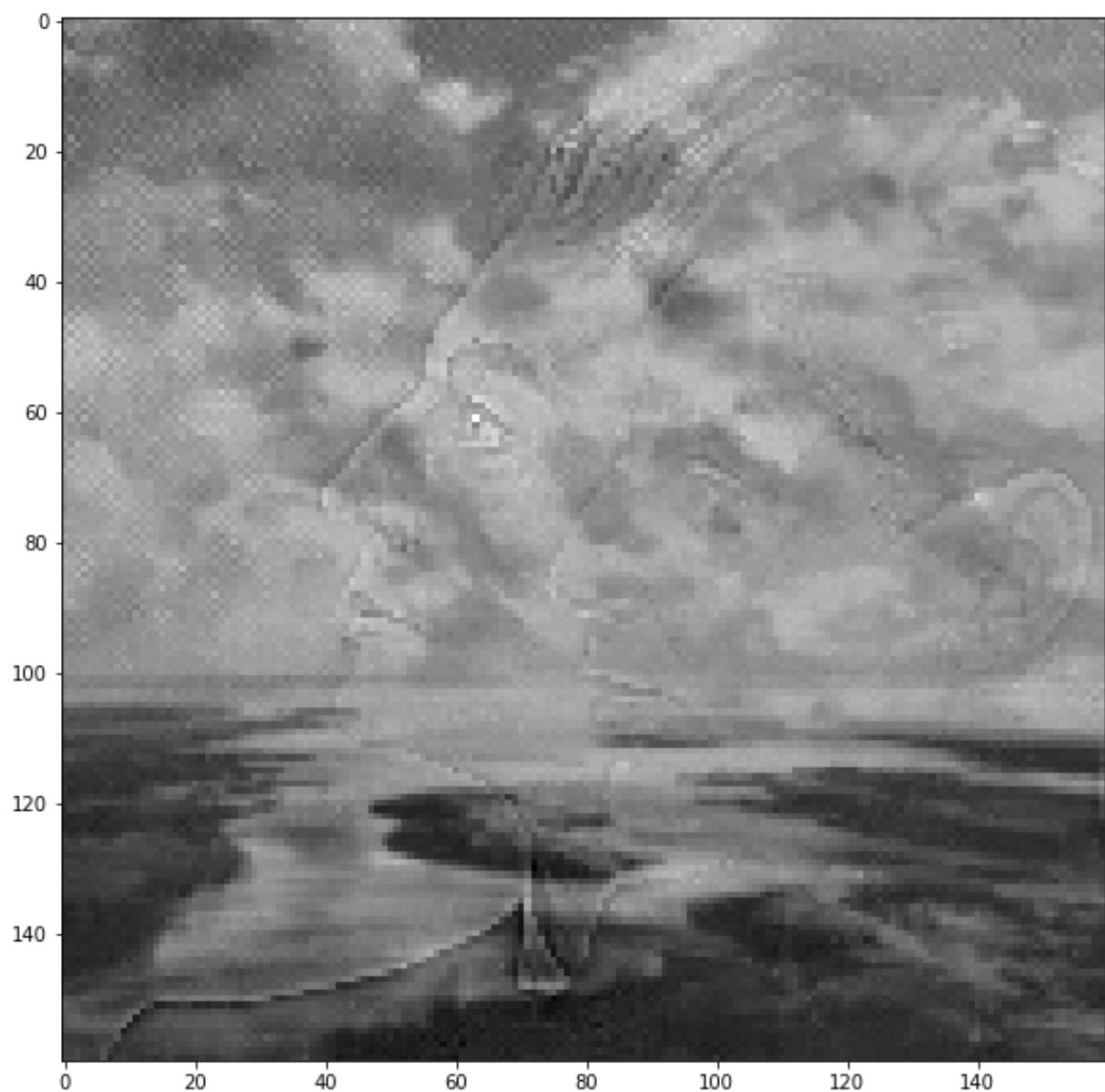




WARNING:root:Lossy conversion from float64 to uint8. Range [-100.288 68711979024, 124.06166344249681]. Convert image to uint8 prior to saving to suppress this warning.



Out[91]: True



In []: