1. Greyscale Reduction
2. Noise Reduction
3. Horinzontal and Vertical Sobel filtering
4. Magnitude Calculation
5. Thresholding

In [5]:
```
import cv2
import numpy as np
import copy
import matplotlib.pyplot as plt
```

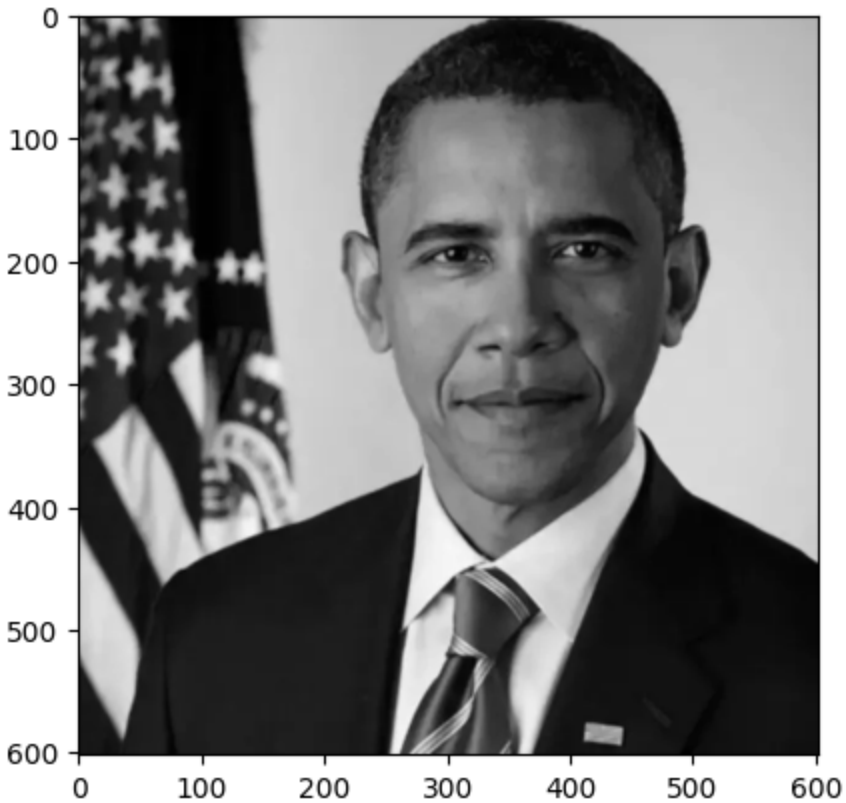
1. Greyscale Reduction

In [6]:
```
# import image
img = cv2.imread("obama.webp")

# Convert to greyscale
img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

plt.imshow(img, cmap="gray")
```

Out[6]:
```
<matplotlib.image.AxesImage at 0x7f6ce808dd50>
```



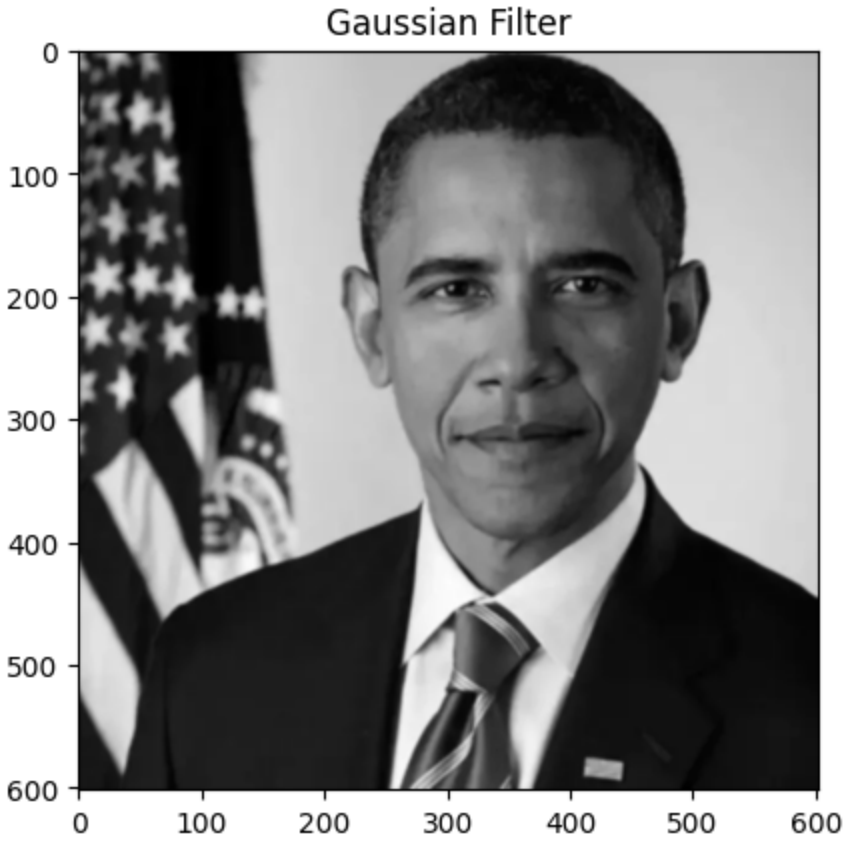
2. Gaussian Noise Reduction

In [7]:
```
# Gaussian filter
gaussian_filter = np.array([[1,2,1],[2,4,2],[1,2,1]], dtype=np.float32) / 16

# Convolve with the image
imgGaussian = cv2.filter2D(img, -1, gaussian_filter)

# Guassian Filter print
plt.imshow(imgGaussian, cmap='gray')
plt.title('Gaussian Filter')
```

Out[7]:
```
Text(0.5, 1.0, 'Gaussian Filter')
```



3. Horinzontal and Vertical Sobel Filtering

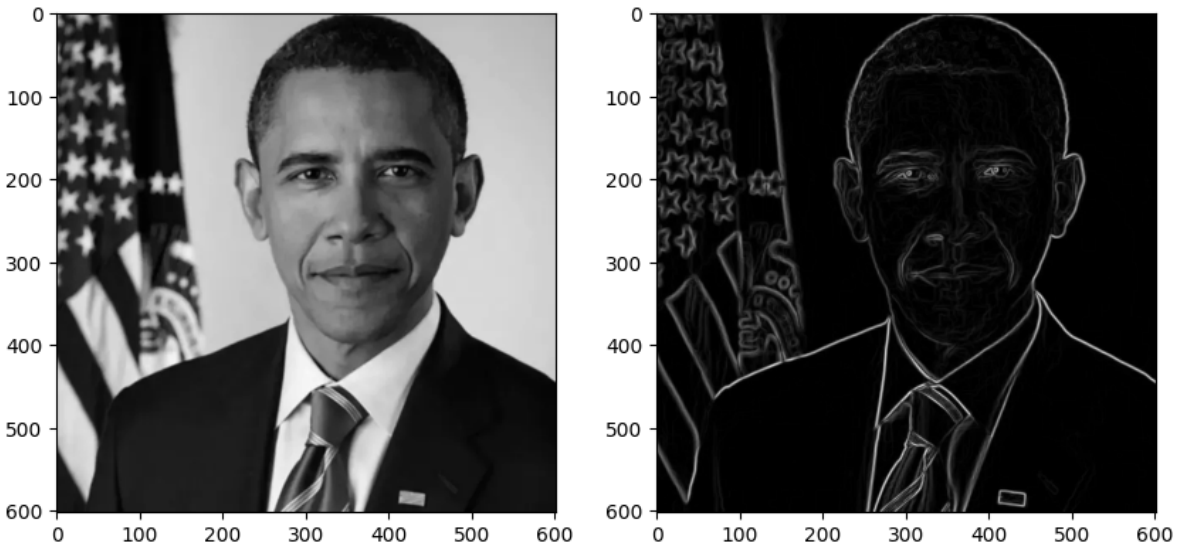
In [8]:
```
# Sobel kernels
kernelX = np.array([[1,2,1],[0,0,0],[-1,-2,-1]], dtype=np.float32)
kernelY = np.array([[-1,0,1],[-2,0,2],[-1,0,1]], dtype=np.float32)

# Sobel convolution
imgSobelX = cv2.filter2D(img, cv2.CV_64F, kernelX)
imgSobelY = cv2.filter2D(img, cv2.CV_64F, kernelY)
```

4. Magnitude Calculation

In [9]:
```
Sobel = cv2.magnitude(imgSobelX, imgSobelY)
fig = plt.figure(figsize = (10,10))
# Left image
fig.add_subplot(1,2,1)
plt.imshow(img, cmap='gray')
# Right image
fig.add_subplot(1,2,2)
plt.imshow(Sobel, cmap='gray')
```

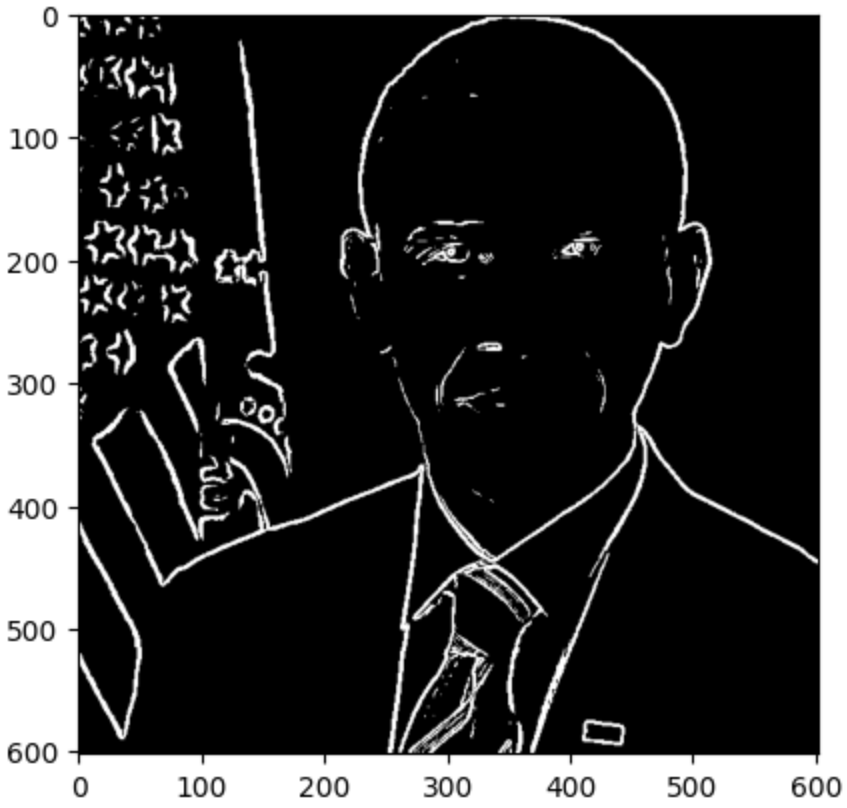
Out[9]:
```
<matplotlib.image.AxesImage at 0x7f6ce1736f50>
```



5. Thresholding

In [18]:
```
ret,thresh = cv2.threshold(Sobel,160,255,cv2.THRESH_BINARY)
plt.imshow(thresh, cmap="gray")
```

Out[18]:
```
<matplotlib.image.AxesImage at 0x7f6ce0573ee0>
```



In [ ]: