

```
In [11]: import cv2
import numpy as np
import matplotlib.pyplot as plt
from io import BytesIO
from PIL import Image
from google.colab.patches import cv2_imshow
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell

has been executed in the current browser session. Please rerun this cell to enable.

```
In [12]: img1 = cv2.imread('cameraman.tif')
cv2_imshow(img1) #Show Image
```



```
In [13]: N1 = 255 - img1
cv2_imshow(N1)
```



```
In [14]: img2 = cv2.imread('lena.tiff')
cv2_imshow(img2)
```



```
In [15]: N2 = 255 - img2  
cv2_imshow(N2)
```



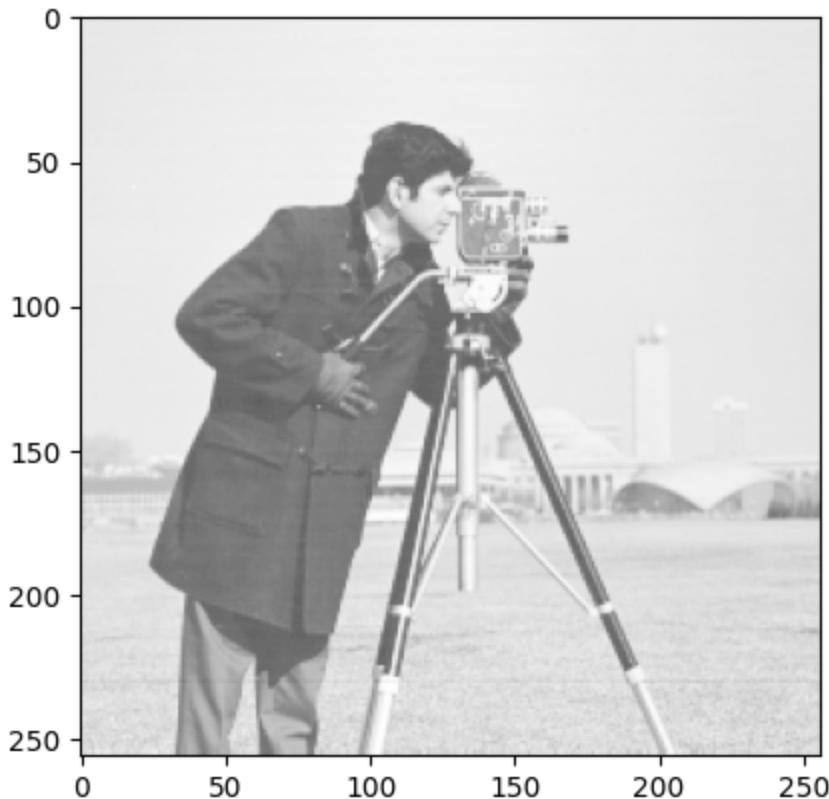
```
In [16]: c = 255 / np.log(1 + np.max(img1))
c
```

```
Out[16]: 46.05103967062828
```

```
In [17]: log_image = c * (np.log(img1 + 1))
log_image = np.array(log_image, dtype=np.uint8)
log_image
```

```
Out[17]: array([[[232, 232, 232],  
                 [233, 233, 233],  
                 [233, 233, 233],  
                 ...,  
                 [231, 231, 231],  
                 [231, 231, 231],  
                 [231, 231, 231]],  
  
                [[234, 234, 234],  
                 [232, 232, 232],  
                 [233, 233, 233],  
                 ...,  
                 [232, 232, 232],  
                 [232, 232, 232],  
                 [231, 231, 231]],  
  
                [[232, 232, 232],  
                 [233, 233, 233],  
                 [233, 233, 233],  
                 ...,  
                 [231, 231, 231],  
                 [231, 231, 231],  
                 [231, 231, 231]],  
  
                ...,  
  
                [[218, 218, 218],  
                 [225, 225, 225],  
                 [222, 222, 222],  
                 ...,  
                 [226, 226, 226],  
                 [226, 226, 226],  
                 [218, 218, 218]],  
  
                [[221, 221, 221],  
                 [223, 223, 223],  
                 [224, 224, 224],  
                 ...,  
                 [225, 225, 225],  
                 [224, 224, 224],  
                 [218, 218, 218]],  
  
                [[221, 221, 221],  
                 [223, 223, 223],  
                 [224, 224, 224],  
                 ...,  
                 [225, 225, 225],  
                 [224, 224, 224],  
                 [218, 218, 218]]], dtype=uint8)
```

```
In [18]: plt.imshow(log_image)  
plt.show()  
cv2_imshow(log_image)
```



```
In [20]: gama = 0.5  
c = 255 / np.max(img1) ** (gama)  
p1_img1 = c * img1  
p1_img1 = c * pow(img1, gama)
```

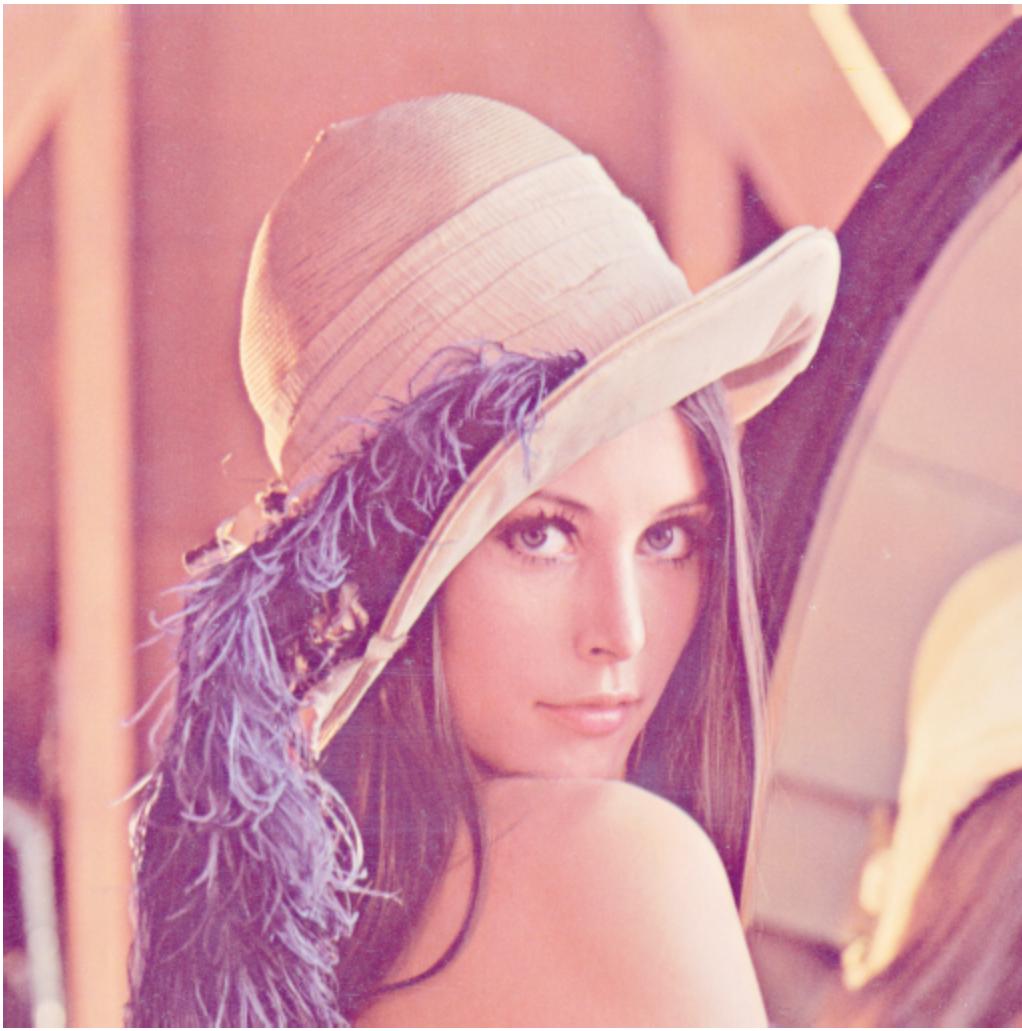
```
In [21]: cv2_imshow(img1)  
cv2_imshow(p1_img1)
```



```
In [22]: gama = 0.5
c = 255 / np.max(img2) ** (gama)
p1_img2 = c * img2
p1_img2 = c * pow(img2, gama)
```

```
In [23]: cv2_imshow(img2)
cv2_imshow(p1_img2)
```



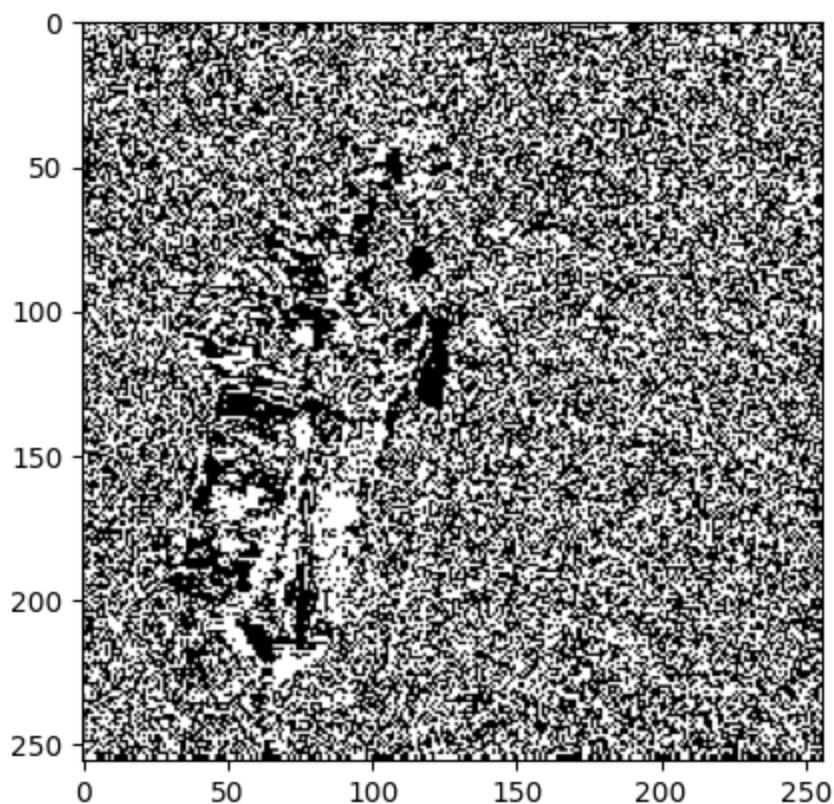
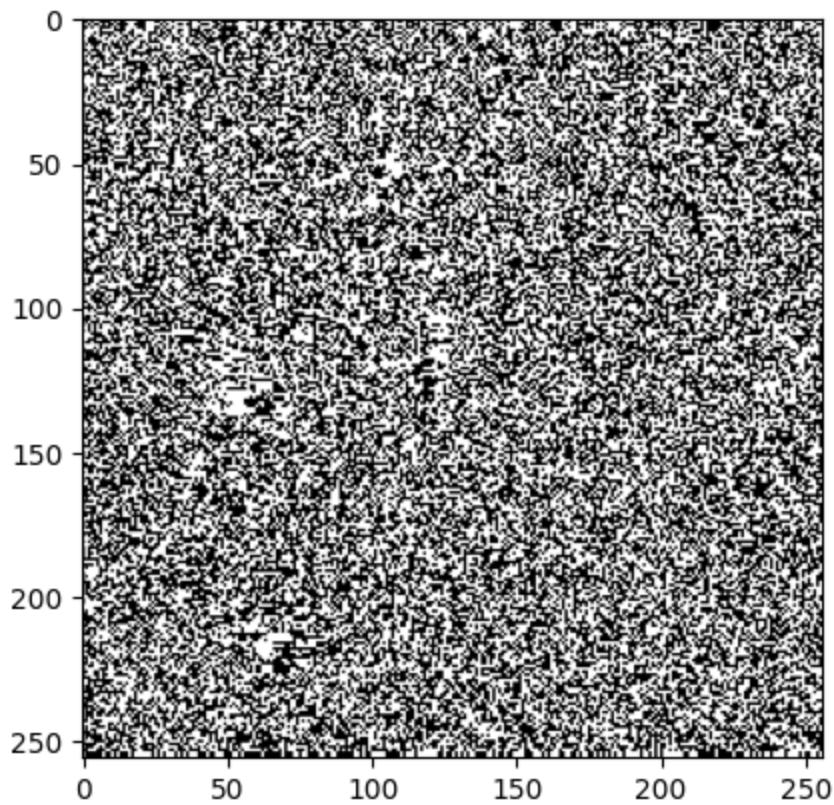


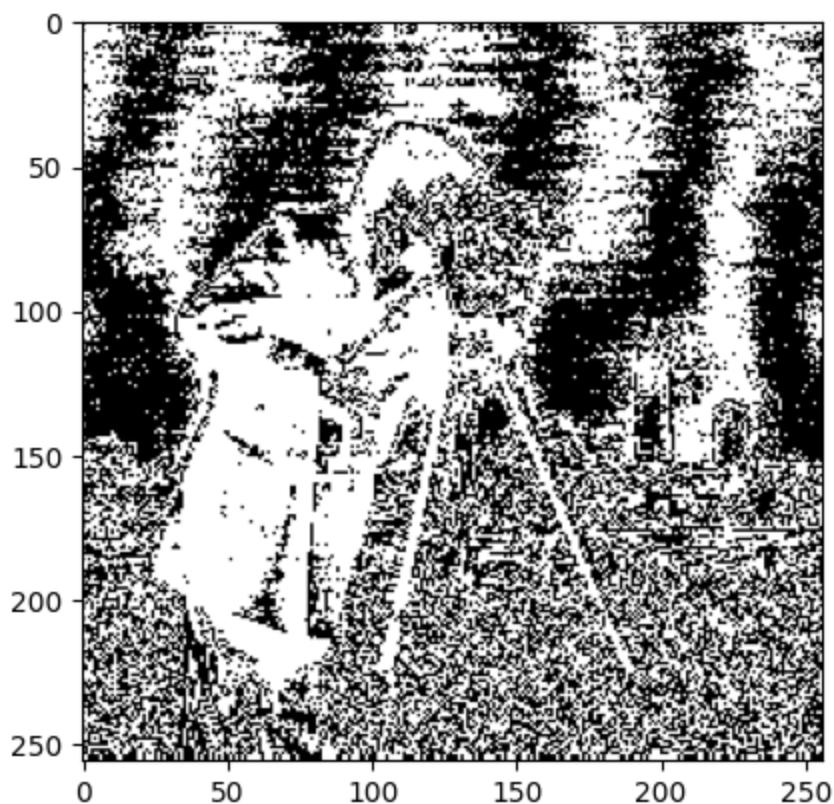
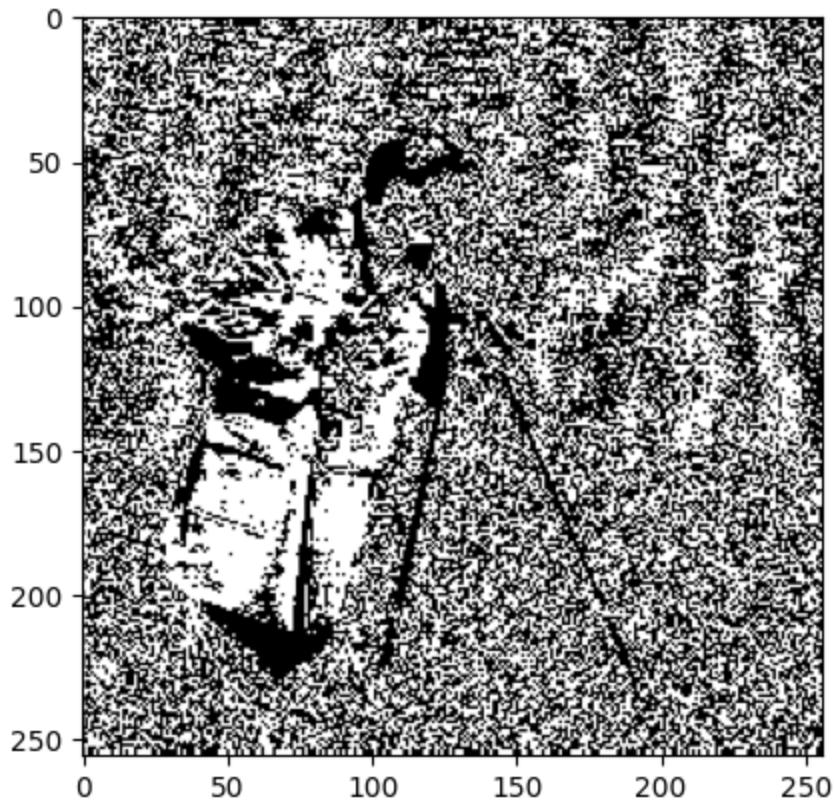
```
In [24]: gray_img = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
[rows,cols] = gray_img.shape;
SL=gray_img;
alpha=0.5
beta=1.2
gamma=0.2
r1=150
r2=200
s1=alpha*r1
s2=beta*(r2-r1)+s1
```

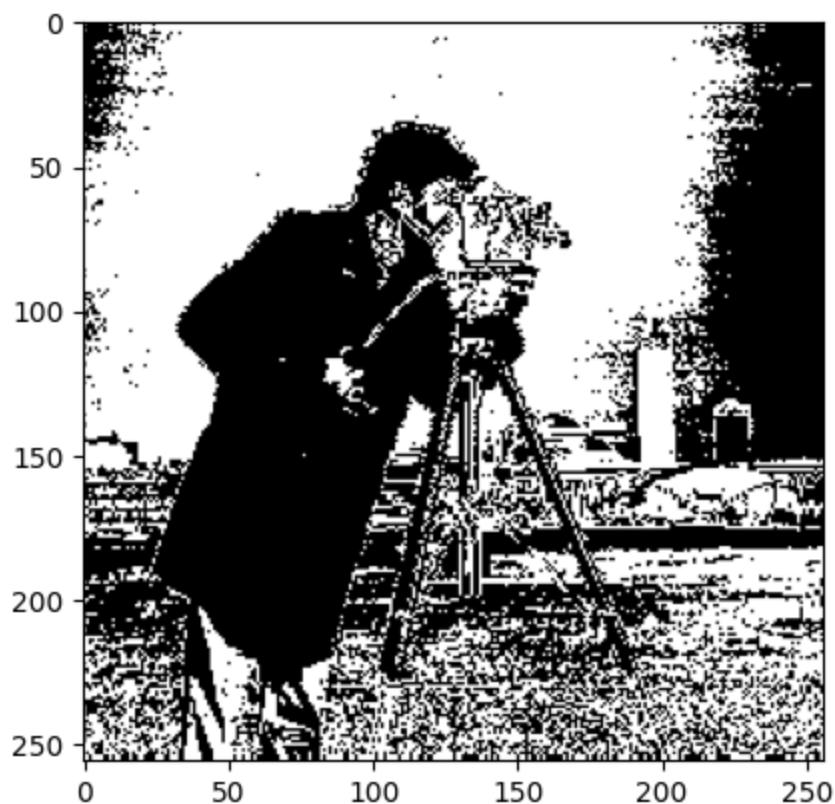
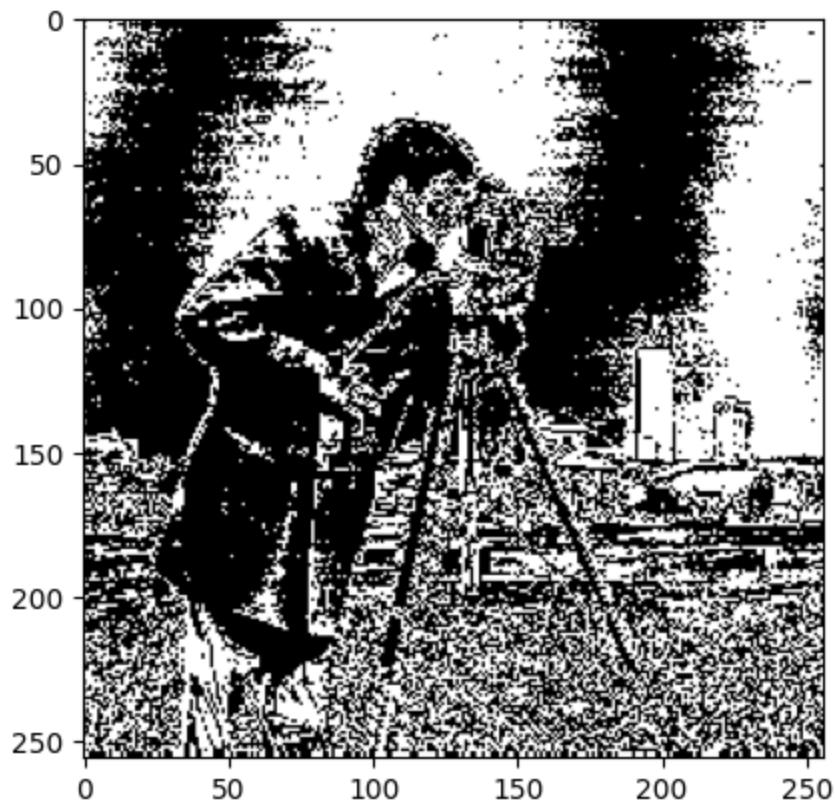
```
In [25]: for r in range(0, rows-1):
    for c in range(0, cols-1):
        if (gray_img[r,c]<r1):
            SL[r,c]=alpha*gray_img[r,c];
        else:
            if (gray_img[r,c]<r2):
                SL[r,c]=beta*(gray_img[r,c]-r1)+s1;
            else:
                SL[r,c]=gamma*(gray_img[r,c]-r2)+s2;
cv2_imshow(gray_img)
cv2_imshow(SL)
```

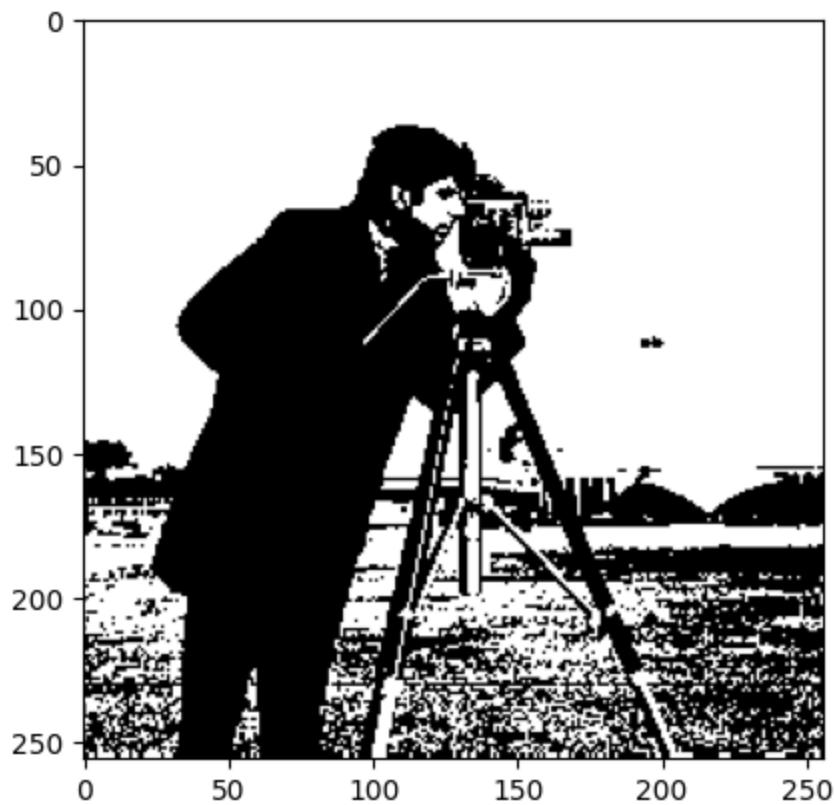
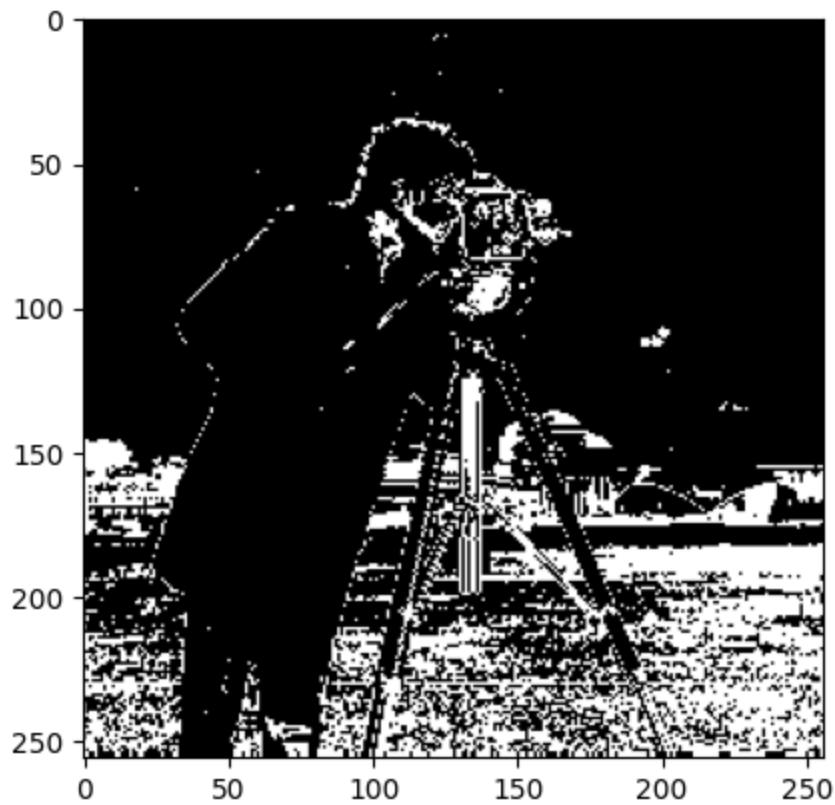


```
In [26]: from matplotlib import pyplot as plt3
imgs = [255 * ((img1& (1<<i)) >>i) for i in range(8)]
for i in range(8):
    plt3.imshow(imgs[i],cmap='gray')
    plt3.show()
cv2.imwrite('bsp_1.jpg', imgs[0])
cv2.imwrite('bsp_2.jpg', imgs[1])
cv2.imwrite('bsp_3.jpg', imgs[2])
cv2.imwrite('bsp_4.jpg', imgs[3])
cv2.imwrite('bsp_5.jpg', imgs[4])
cv2.imwrite('bsp_6.jpg', imgs[5])
cv2.imwrite('bsp_7.jpg', imgs[6])
cv2.imwrite('bsp_8.jpg', imgs[7])
```









Out[26]: True