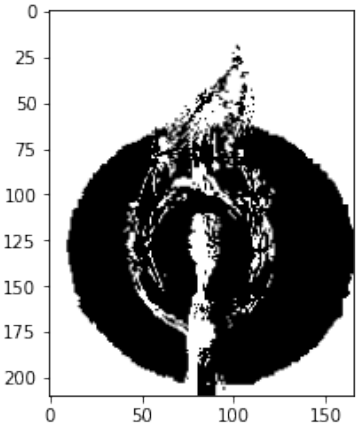
[9]:
```
import numpy as np
import cv2, matplotlib.pyplot as plt
%matplotlib inline
```

[18]:
```
# read image and show
img = cv2.imread('5d.jpg') # in BGR mode
# convert to RGB mode
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.axis('off')
plt.imshow(img)
```

[11]:
```
# performing otsu's binarization and convert to gray scale first
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.
 ↪THRESH_OTSU)
print("Threshold limit: " + str(ret))

plt.axis('on')
plt.imshow(thresh, cmap = 'gray')
```

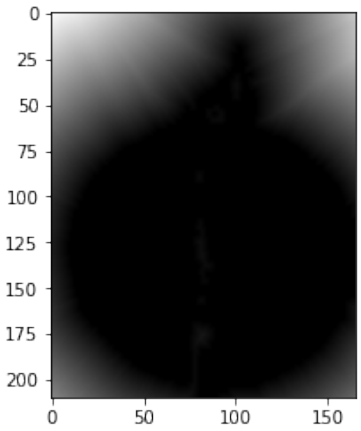
```
Threshold limit: 79.0
```



[12]:
```
# noise removal
kernel = np.ones((3, 3), np.uint8)
opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel, iterations = 2)
# sure background area
sure_bg = cv2.dilate(opening, kernel, iterations = 3)
# sure foreground area
dist_transform = cv2.distanceTransform(opening,cv2.DIST_L2,5)
ret, sure_fg = cv2.threshold(dist_transform,0.7*dist_transform.max(),255,0)

# Finding unknown region
sure_fg = np.uint8(sure_fg)
unknown = cv2.subtract(sure_bg,sure_fg)
```

[13]:
```
plt.imshow(dist_transform, cmap = 'gray')
```

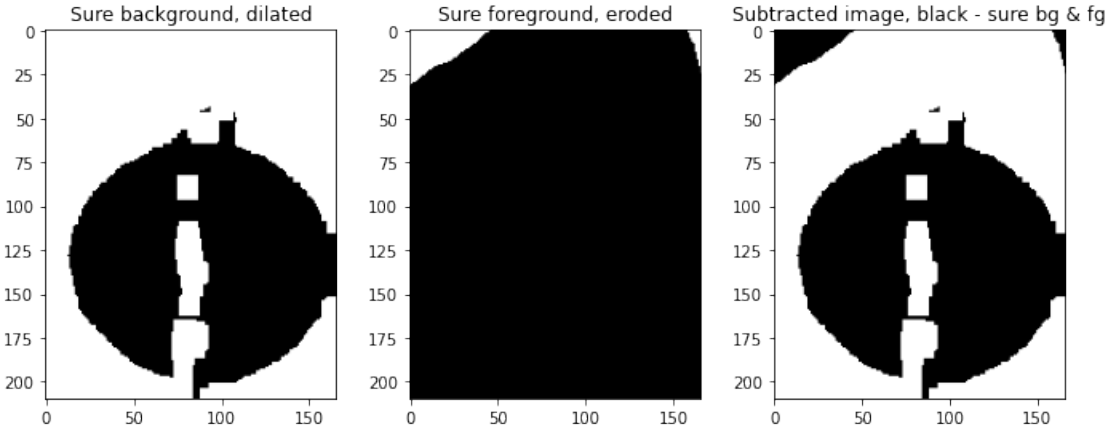


[15]:
```
fig = plt.figure(figsize = (10, 5)) # to change figsize
plt.subplot(131)
plt.imshow(sure_bg, cmap = 'gray')
plt.title('Sure background, dilated')

plt.subplot(132)
plt.imshow(sure_fg, cmap = 'gray')
plt.title('Sure foreground, eroded')

plt.subplot(133)
plt.imshow(unknown, cmap = 'gray')
plt.title('Subtracted image, black - sure bg & fg')
plt.tight_layout()

# plt.subplots_adjust(wspace = 3)
# fine tuning
# f.subplots_adjust(wspace=3)
```


Sure background, dilated
Sure foreground, eroded
Subtracted image, black - sure bg & fg
0
25
50
75
100
125
150
175
200
0
50
100
150