Implementar 2: Binarización local, método Bernsen

Implementar para una ventana de 3x3, 5x5 o 7x7.

Contraste: Máximo-Mínimo

gris_medio: Media

In [1]:
```
%matplotlib inline

import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
```

In [2]:
```
def get_window(image, center_x, center_y, size):
    image_height, image_width = image.shape
    window_half_size = np.floor(size/2).astype(int)

    top = max(center_y - window_half_size, 0)
    bottom = min(center_y + window_half_size, image_height - 1)
    left = max(center_x - window_half_size, 0)
    right = min(center_x + window_half_size, image_width - 1)

    return image[top:bottom, left:right]
```

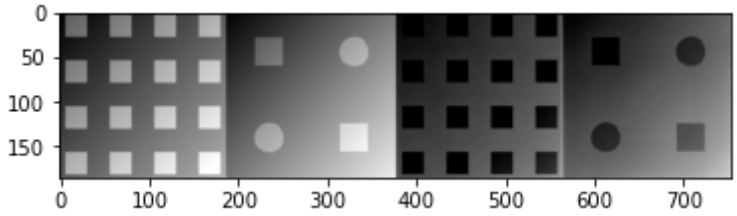
In [3]:
```
#if(contraste_local < contraste_referencia)
#  pixel=(gris_medio >= 128)? objeto:background
#else
#  pixel=(pixel >= gris_medio)? objeto:background

def bernsen(image, contrast, window_size):
    rows = image.shape[0]
    columns = image.shape[1]
    output = image.copy()

    for i in range(rows):
        for j in range(columns):
            window = get_window(image, j, i, window_size)
            local_contrast = np.max(window) - np.min(window)
            mean = np.mean(window)
            if local_contrast < contrast:
                if mean >= 128:
                    output[i, j] = 255
                else:
                    output[i, j] = 0
            else:
                if output[i, j] >= mean:
                    output[i, j] = 255
                else:
                    output[i, j] = 0
    return output
```

In [4]:
```
original_image = cv.imread("Sombreado.png", cv.IMREAD_GRAYSCALE)
window_sizes = (3, 5, 7, 9, 11, 13, 15)
contrast = 2
```

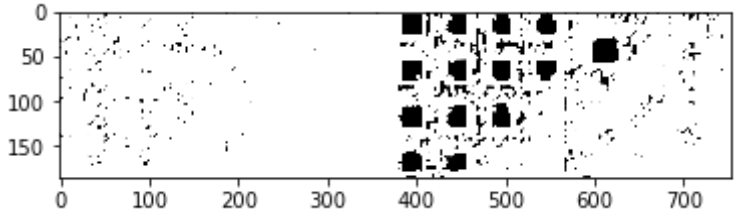
In [5]:
```
print("Original")
plt.imshow(original_image, cmap='gray', vmin=0, vmax=255)
```

```
plt.show()

for window_size in window_sizes:
    print("Processing Bernsen with window size {0}X{0}...\n".format(window_si
    binarized_image = bernsen(original_image, contrast, window_size)
    print("Binarized - Bernsen {0}X{0}".format(window_size))
    plt.figure(window_size)
    plt.imshow(binarized_image, cmap='gray', vmin=0, vmax=1)
    plt.show()
```

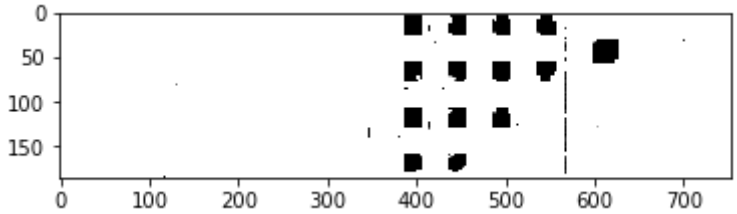
```
Original
```



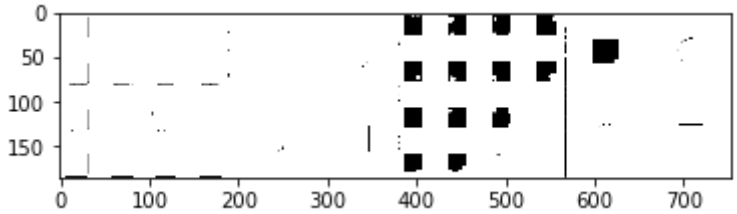

```
Processing Bernsen with window size 3X3...

Binarized - Bernsen 3X3
```



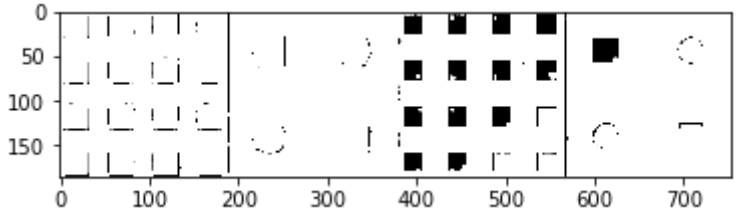

```
Processing Bernsen with window size 5X5...

Binarized - Bernsen 5X5
```



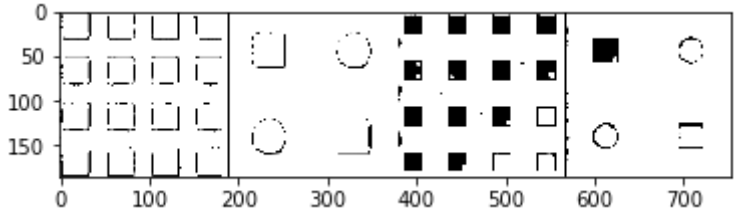

```
Processing Bernsen with window size 7X7...

Binarized - Bernsen 7X7
```




```
Processing Bernsen with window size 9X9...

Binarized - Bernsen 9X9
```

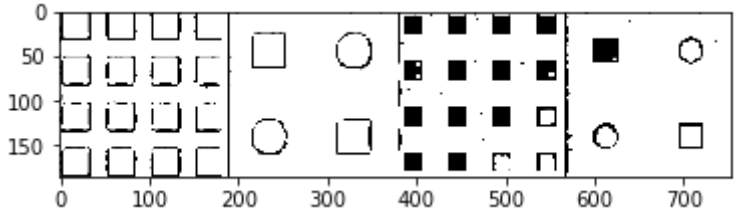



```
Processing Bernsen with window size 11X11...

Binarized - Bernsen 11X11
```



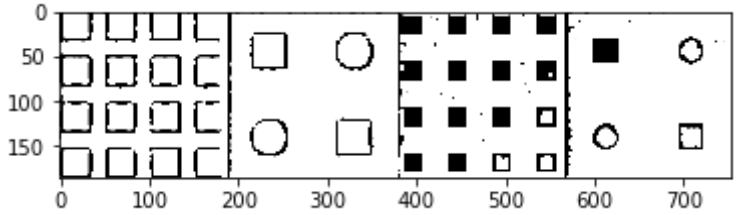
```
Processing Bernsen with window size 13X13...
```

```
Binarized - Bernsen 13X13
```



```
Processing Bernsen with window size 15X15...
```

```
Binarized - Bernsen 15X15
```



```
In [ ]:
```