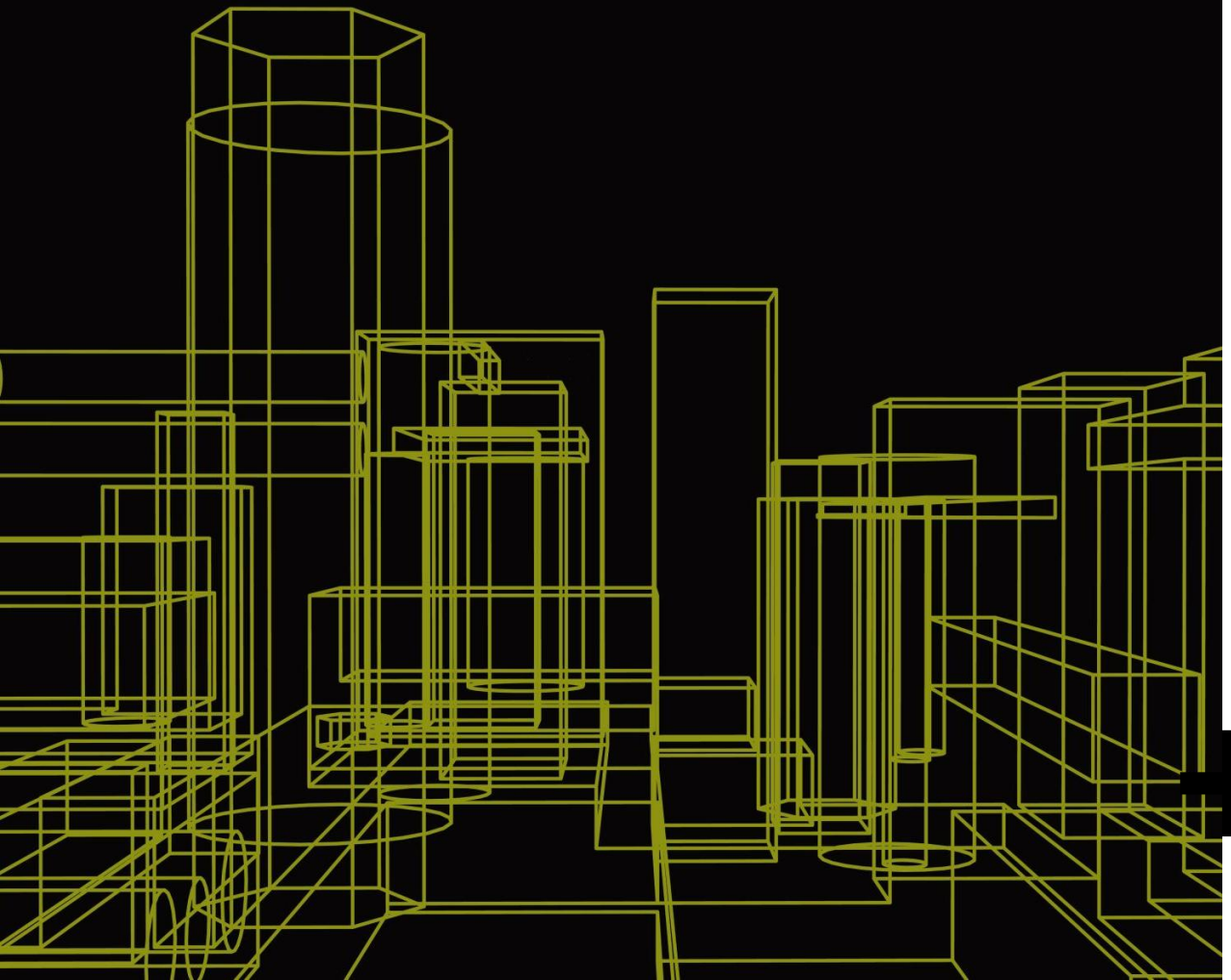


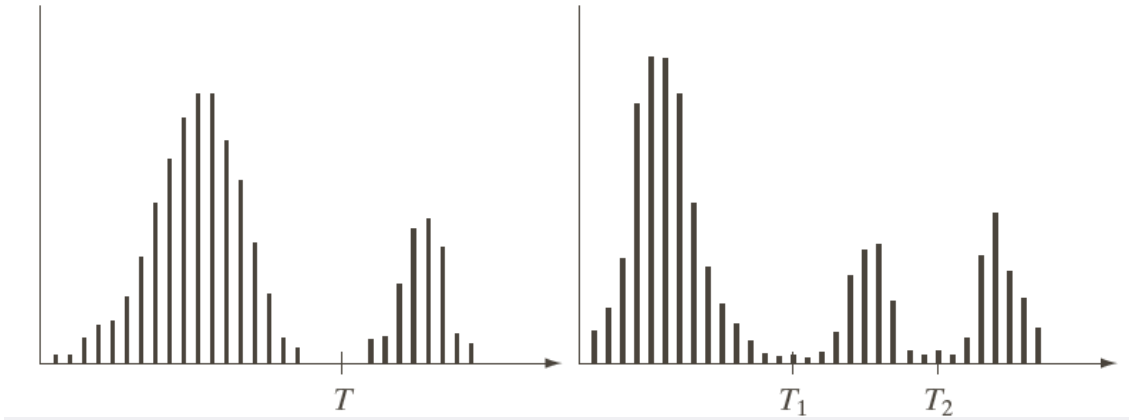
Image Segmentation

Thresholding

Basic Process

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases}$$

$$g(x, y) = \begin{cases} a & \text{if } f(x, y) > T_2 \\ b & \text{if } T_1 < f(x, y) \leq T_2 \\ c & \text{if } f(x, y) \leq T_1 \end{cases}$$



Basic Thresholding

- import cv2
- import numpy as np
- import matplotlib.pyplot as plt

- # Load the input image in grayscale
- image = cv2.imread('input_image.jpg', cv2.IMREAD_GRAYSCALE)

- # Define the threshold value
- threshold_value = 127
- max_value = 255

- # Create an empty array for the output image
- thresholded_image = np.zeros_like(image)

```
# Apply global thresholding algorithm manually
for i in range(image.shape[0]):
    for j in range(image.shape[1]):
        if image[i, j] > threshold_value:
            thresholded_image[i, j] = max_value
        else:
            thresholded_image[i, j] = 0

# Display the input and output images
plt.figure(figsize=(10, 5))

# Input Image
plt.subplot(1, 2, 1)
plt.title("Input Image")
plt.imshow(image, cmap='gray')
plt.axis('off')

# Thresholded Image
plt.subplot(1, 2, 2)
plt.title(f"Thresholded Image (T={threshold_value})")
plt.imshow(thresholded_image, cmap='gray')
plt.axis('off')

plt.tight_layout()
plt.show()
```

Key factors

– Separation between peaks

– Noise content in image

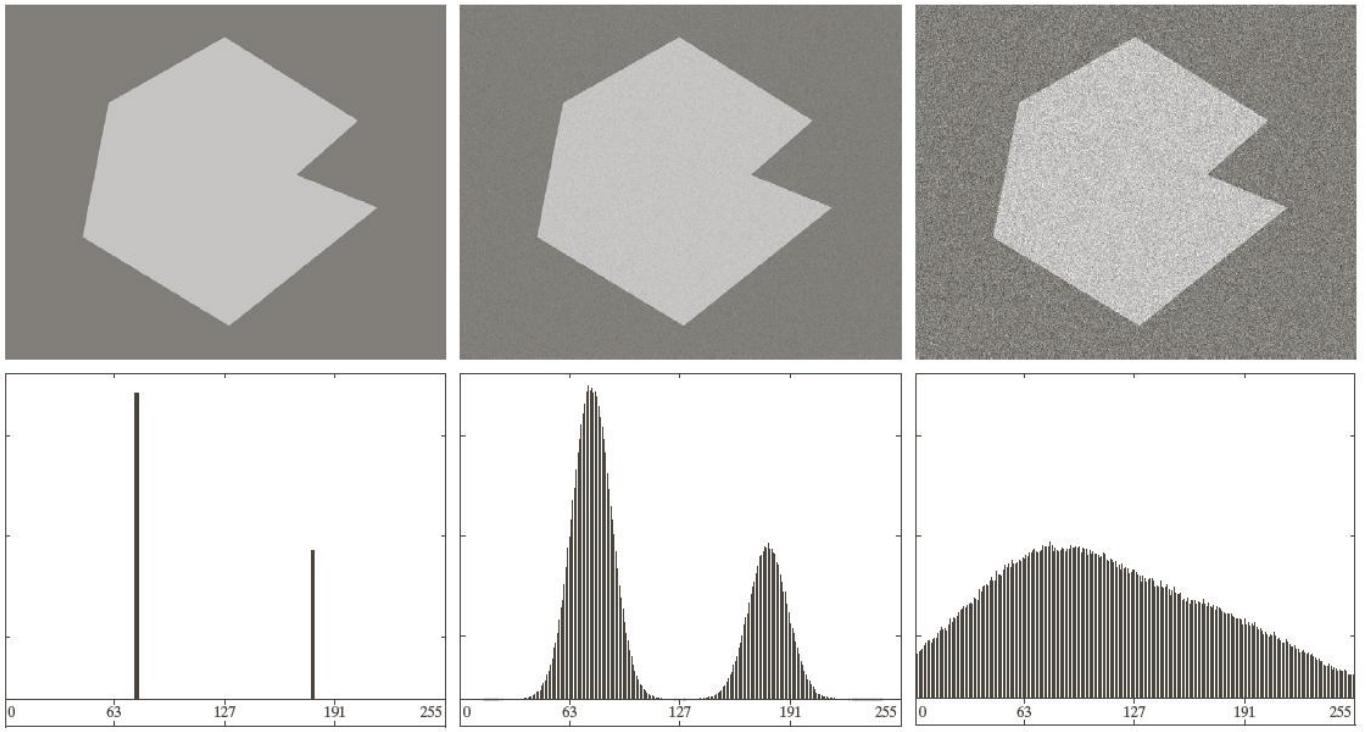
– Relative size of objects and background

– Uniformity of illumination source

– Uniformity of reflectance properties

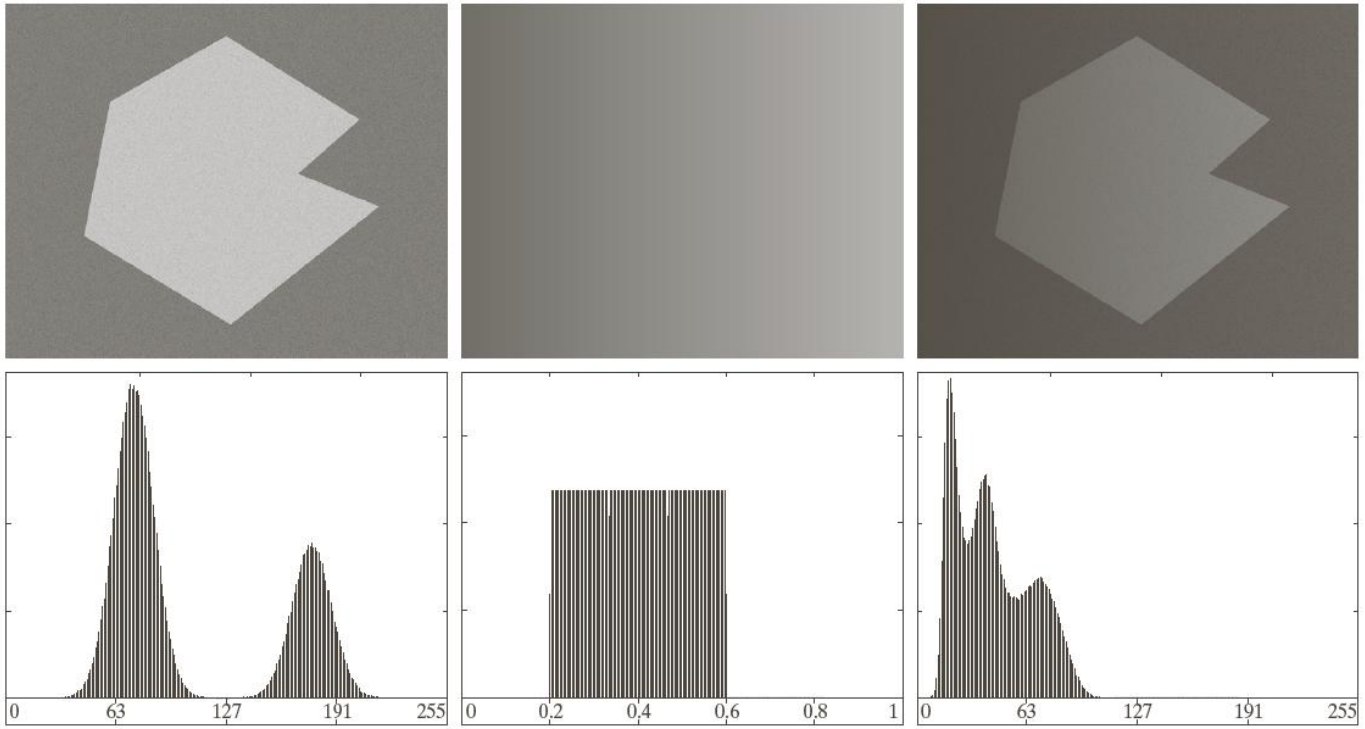
Effect of Noise

- Gaussian noise of zero mean and standard deviation of 10 intensity levels
- Gaussian noise of zero mean and standard deviation of 50 intensity levels
- Apply filters before Segmentation



Effect of illumination and reflectance

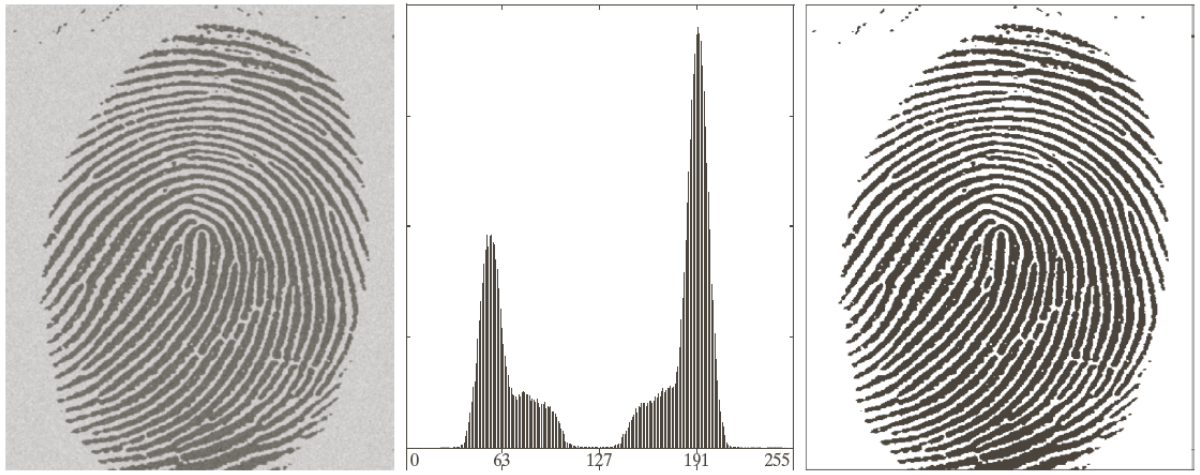
- Correct the shading pattern directly
- Correct the Global shading pattern via pre-processing like top-hat transformation
- Variable thresholding



Basic Global Thresholding

1. Select an initial estimate for the global threshold, T.
2. Segment the image using T.
    1. G_1 consisting of all pixels with intensity values > T
    2. G_2 consisting of all pixels with intensity values ≤ T
3. Compute the average intensity values m_1 and m_2 for the pixels in G_1 and G_2, respectively.
4. Compute a new threshold value $T = ½ (m_1 + m_2)$
5. Repeat steps 2 through 4 until the difference between values of T in successive iterations is smaller than a predefined parameter ΔT.

Basic Global Thresholding



+