Image Segmentation

Edge Detection

- Edge detection is the most common approach for detecting meaningful discontinuities in intensity values.
- Such discontinuities can be determined by using first and second order derivatives (as explained before).
- The 1st-order derivative of an image f(x,y) is defined as:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}, \text{ its magnitude is } \nabla f = \sqrt{G_x^2 + G_y^2} \approx G_x^2 + G_y^2,$$

and its direction is defind as: $\alpha(x,y) = \tan^{-1}(\frac{G_x}{G_y})$.

- The second-order derivative, is calculated by the Laplacian, but it is not used for edge detection due to its sesitivity to noise.
- Different Edge detection schemes can be desigened depending on how Gx and Gy are modelled.

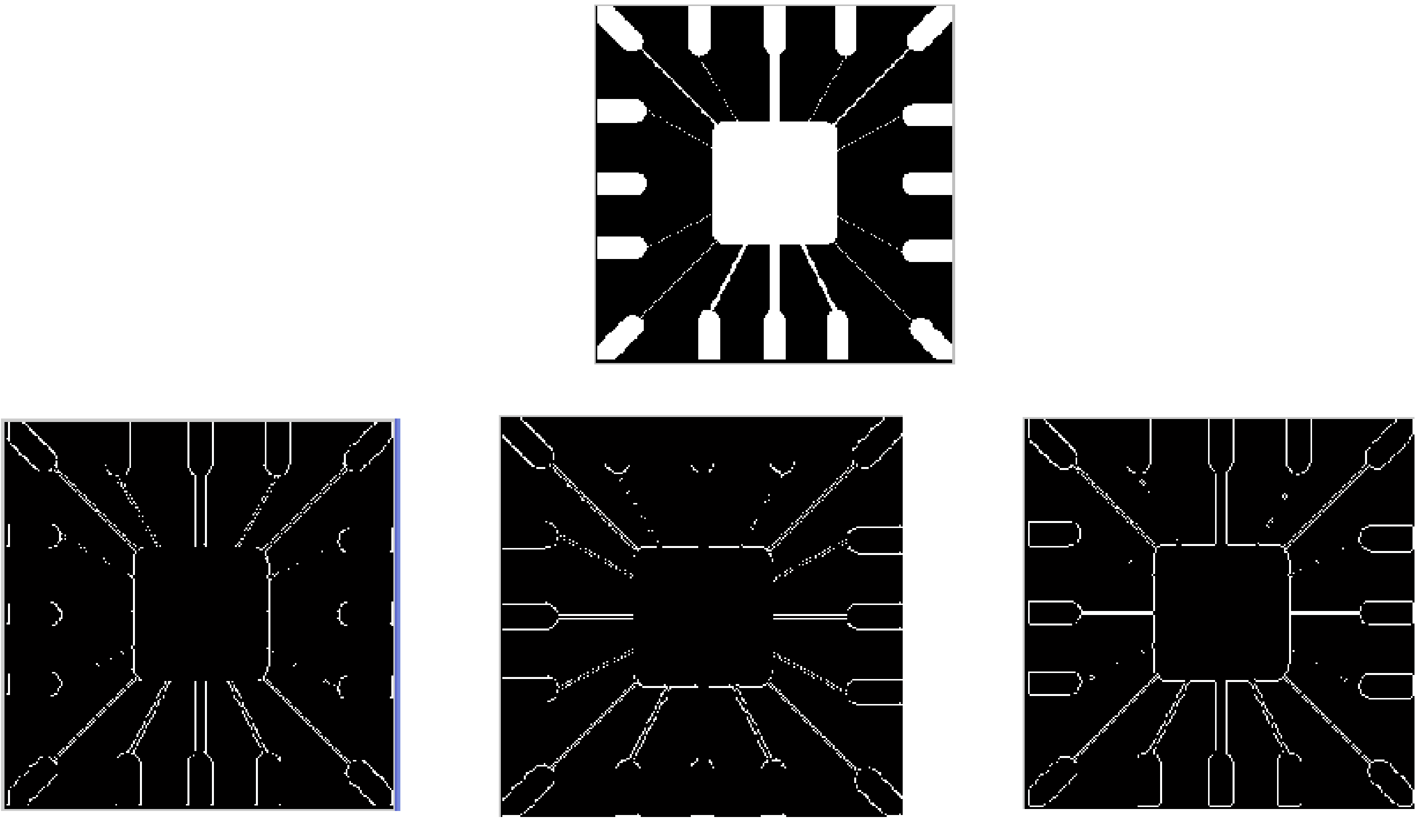
The Sobel method approximates G_x and G_y by the filters:

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \text{ and } G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.$$

The Prewitt method approximates G_x and G_y by the filters:

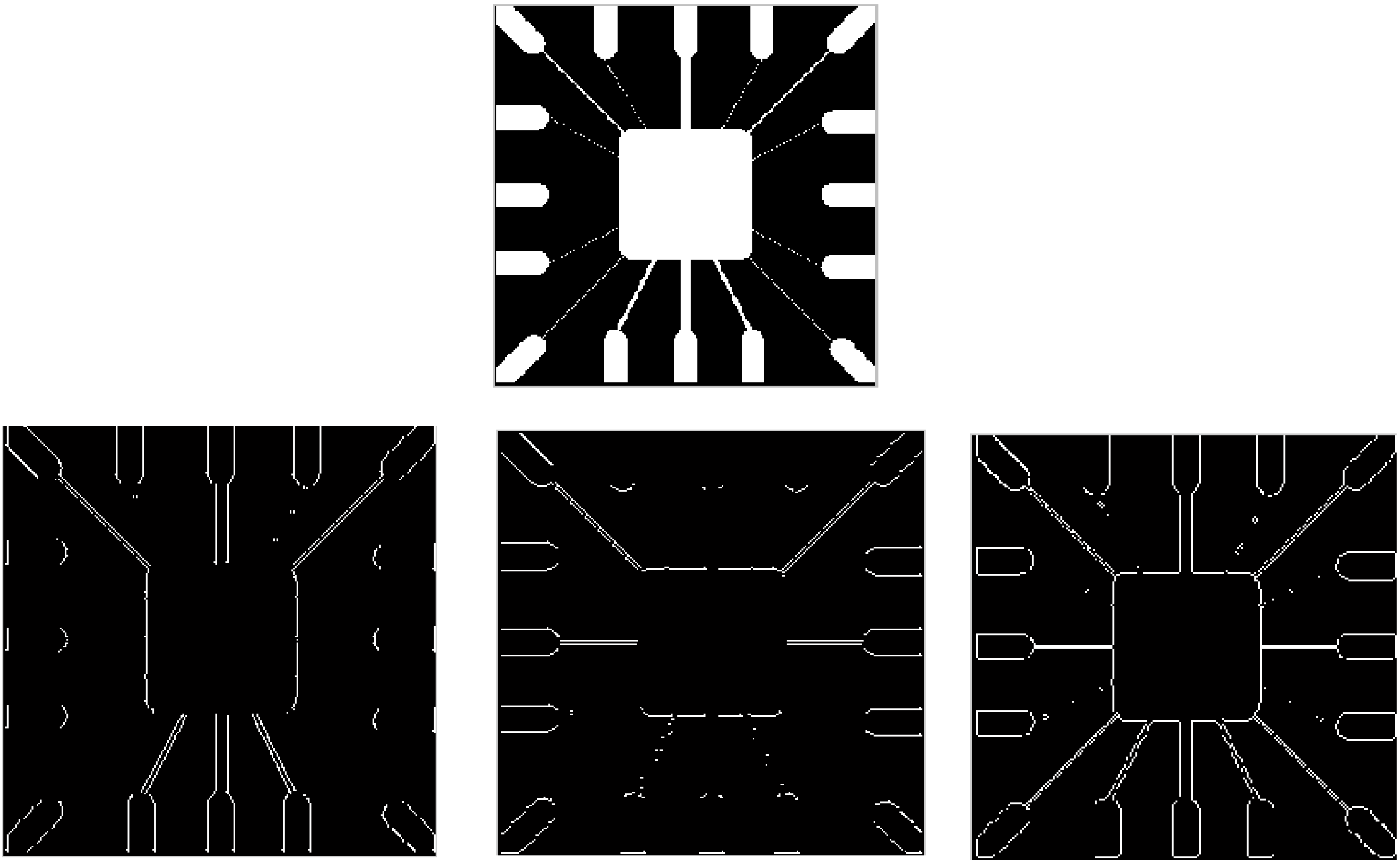
$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \text{ and } G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}.$$

- **Sobel edge detection example.**
- Source image, vertical, horizontal, both.



Prewitt edge detection example.

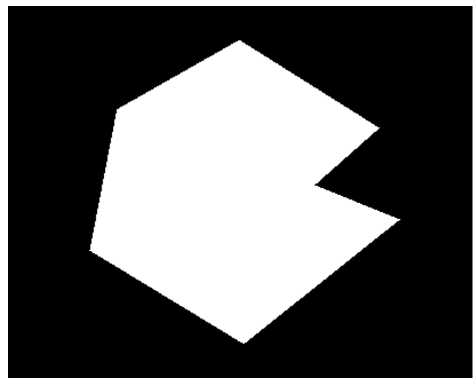
- Source image, vertical, horizontal, both.

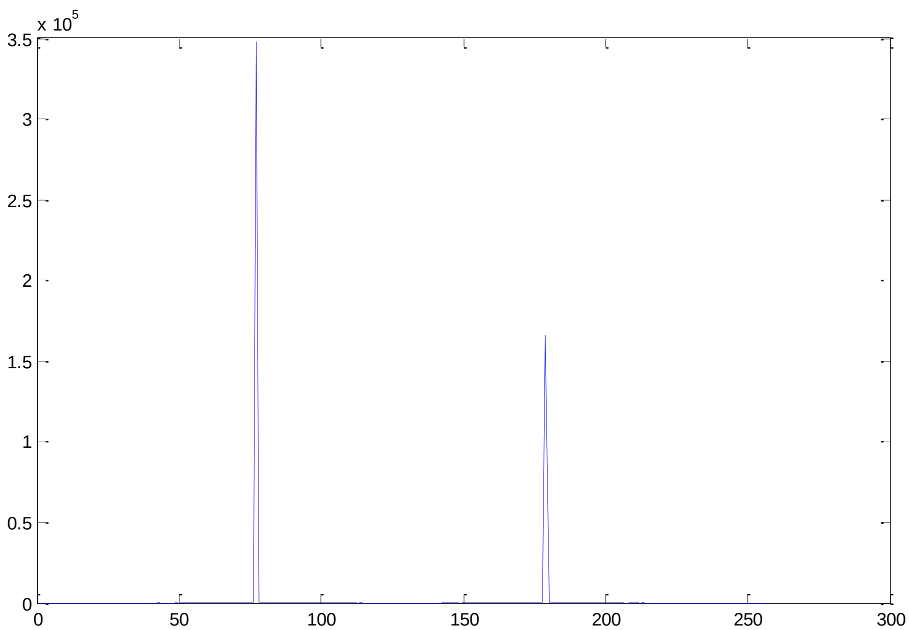


Segmentation using Thresholding

- Because of its simple implementation image thresholding enjoys a central position in applications of image segmentation.
- Thresholding is useful in removing background and highlighting image objects.
- Image histograms are useful in selecting suitable threshold(s), e.g. search for valleys in Histograms.

- If histogram is bi-model (i.e. 2 separable components), a threshold T between the parts yields a segmented image
- When T is a constant, this approach is called global thresholding.
- T can be chosen either by visual inspection or try and error.





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

T = 140.

- If histogram is multi-model (i.e. >2 separable parts), such as when we have 2 light objects, with different intensities, on a dark background then multi-level thresholds (T1, T2) separating the parts works well:

$$g(x,y) = \begin{cases} 127 & \text{if } f(x,y) > T_2 \\ 255 & \text{if } T_1 < f(x,y) \le T_2 \\ 0 & \text{Otherwise} \end{cases} .$$

  - One may use different values than 127 and 255.

Automatic Global Thresholding

- The above method is OK assuming uniform illumination, and is called global thresholding.
- The following is an automatic iterative algorithm to select a global segmenting threshold T:

  1- Select an initial estimate for T (e.g. (Max+Min)/2),

  2- Segment the image using T. Let G_1 be the set of all pixels with gray level > T , and G_2 be the rest.

  3- Compute the average intensity value μ_1 and μ_2 for the pixels in regions G_1 and G_2 .

  4- Compute a new threshold value T= $(\mu_1 + \mu_2)/2$

5- Repeat steps 2 through 4 until the difference between two successive T's is smaller than a selected small value (T_o).

Effect of Illumination on Thresholding

- Variation in illumination makes Thresholding more challenging (Histograms may not have valleys).
- What is needed is Adaptive Thresholding schemes.
- Subdivide the image into blocks and apply different thresholding in the different blocks.