Image processing

Image Quantization
&
Edge Detection

Ch2 – part 4

Image Quantization

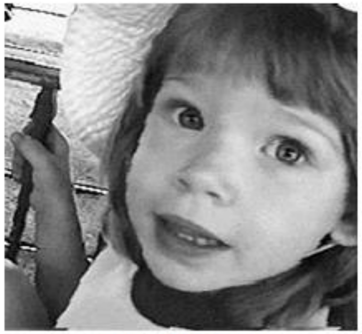
Image quantization is the process of reducing the image data by removing some of the detail information by mapping group of data points to a single point. This can be done by:

1. Gray_Level reduction (reduce pixel values themselves I(r, c).
2. Spatial reduction (reduce the spatial coordinate (r, c).

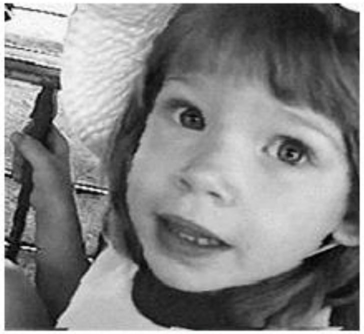
The simplest method of gray-level reduction is Thresholding. We select a threshold gray _level and set everything above that value equal to “1” and everything below the threshold equal to “0”. This effectively turns a gray_level image into a binary (two level) image and is often used as a preprocessing step in the extraction of object features, such as shape, area, or perimeter.

Image Quantization

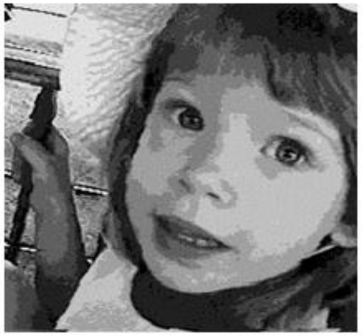
A more versatile method of gray _level reduction is the process of taking the data and reducing the number of bits per pixel. This can be done very efficiency by masking the lower bits via an AND operation. Within this method, the numbers of bits that are masked determine the number of gray levels available.



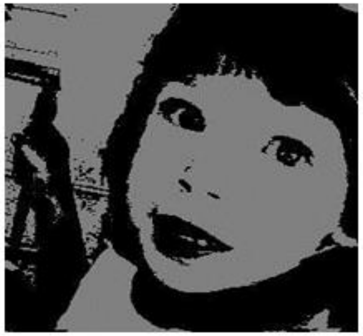
Original 8-bit image,
256 gray levels



Quantized to 6 bits,
64 gray levels



Quantized to 3 bits,
8 gray levels



Quantized to 1 bits,
2 gray levels

Image Quantization

Example: We want to reduce 8_bit information containing 256 possible gray_level values down to 32 possible values.

This can be done by ANDing each 8-bit value with the bit string 11111000. this is equivalent to dividing by eight(2^3), corresponding to the lower three bits that we are masking and then shifting the result left three times. [Gray _level in the image 0-7 are mapped to 0, gray_level in the range 8-15 are mapped to 8 and so on].

We can see that by masking the lower three bits we reduce 256 gray levels to 32 gray levels:

$$\mathbf{256 \div 8 = 32}$$

EX/

$(222)_{10}=(11011110)_2$

ANDing with 11111000

$= (11011000)_2 = (216)_{10}$

Image Quantization

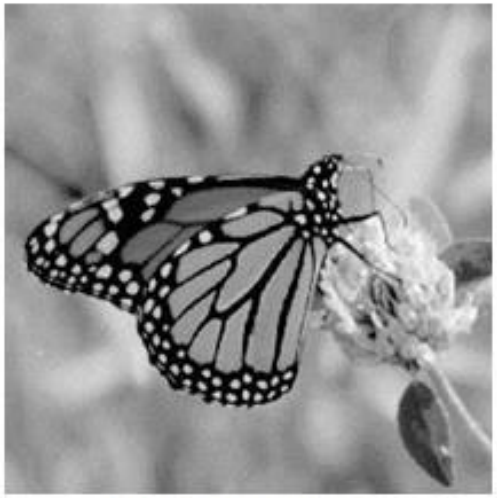
- ❖ Image quantization by masking to 128 gray levels, this can be done by ANDing each 8-bit value with bit string 11111110(2^1).
- ❖ Image quantization by masking to 64 gray_level. This can be done by ANDing each 8-bit value with bit string 11111100(2^2).

As the number of gray levels decreases, we can see increase in a phenomenon called contouring الكفاف.

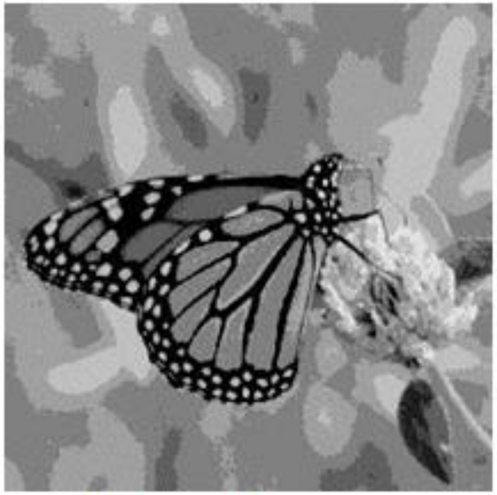
Contouring appears in the image as false edges, or lines as a result of the gray _level quantization method.

Image Quantization

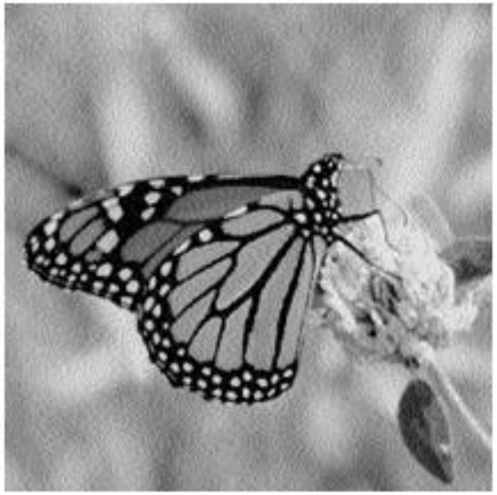
This false contouring effect can be visually improved upon by using an IGS (improved gray-scale) quantization method. In this method (IGS) the improvement will be by adding a small random number to each pixel before quantization, which results in a more visually pleasing appearance.



Original Image



Uniform quantization to 8 levels (3 bits)



IGS quantization to 8 levels (3 bits)

Edge Detection

Detecting edges is a basic operation in image processing. The edges of items in an image hold much of the information in the image.

The edges tell you where:

Items are.

Their size.

shape

and something about their texture.

Edge detection methods are used as a first step in the line detection processes, and they are used to find object boundaries by marking potential edge points corresponding to place in an image where rapid changes in brightness occur. After these edge points have been marked, they can be merged to form lines and objects outlines.

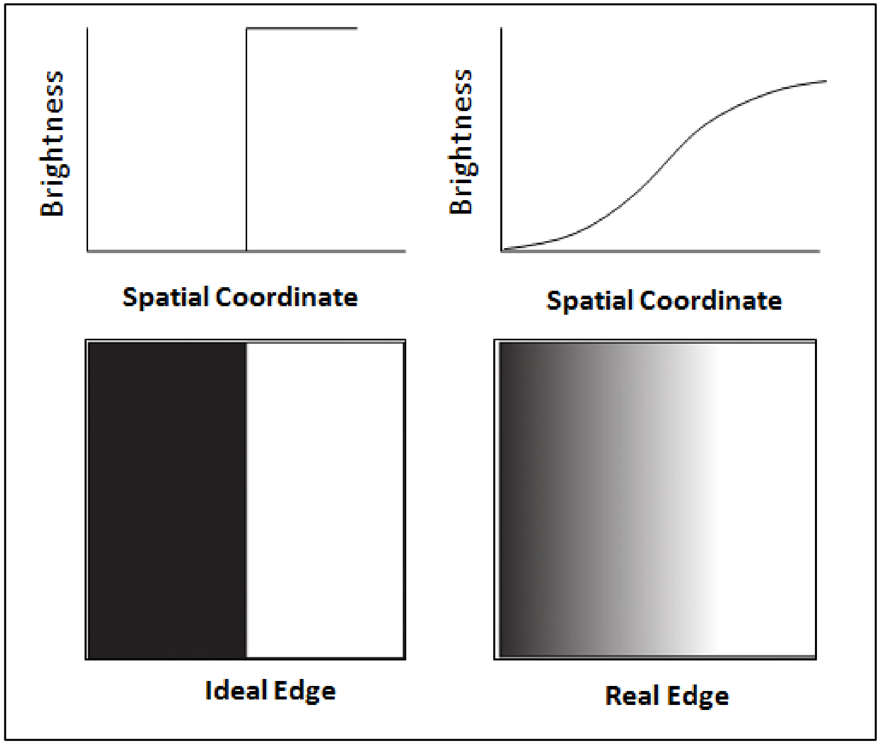
Edge Detection

Edge detection operations are based on the idea that edge information in an image is found by looking at the relationship a pixel has with its neighbors. If a pixel gray_level values similar to those around it, there is probably not an edge at that point. However, if a pixel has neighbors with widely varying gray levels, it may represent an edge point. In other words, an edge is defined by a discontinuity in gray-level values. Ideally, an edge separates two distinct objects. In practice, edges are caused by:

- Change in color or texture or
- Specific lighting conditions present during the image acquisition process.

Edge Detection

The vertical axis represents brightness, and the horizontal axis shows the spatial coordinates. The abrupt change in brightness characterizes an ideal edge. In the previous figure we see the representation of real edge, which change gradually.



The difference between an ideal edge and a real edge

Edge Detection

This gradual change is a minor form of blurring caused by:

- Imaging devices
- The lenses
- Or the lighting and it is typical for real world (as opposed to computer _generated) images.

An edged is where the gray level of the image moves from an area of low values to high values or vice versa. The edge itself is at the center of this transition. The detected edge gives a bright spot at edge and dark area everywhere else. This mean it is the slope or rate of change of the gray level in the edge.

Edge Detection

How do you calculate the derivative (the slop) of an image in all direction?

Convolution of the image with masks is the most often used techniques of doing this.

The idea is to take a 3×3 array of numbers and multiply it point by point with 3×3 section of the image you sum the products and place the result in the center point of the image.

The question in this operation is how to choose the 3×3 mask?

There are several masks that amplify the slop of the edge. Take the simple one-dimensional case and take as an example points on the ideal edge near the edge. They could have values such as [3 5 7]. The slop through these three points is (7-3)/2=2.if you convolve these three point with [-1 0 1] you have -3+7=4.

Edge Detection

The **convolution amplified the slope**, and the result is a large number at the transition point in the edge. There are two basic principles for each edge detector mask:

- **First**: the number in the mask sum to zero. If 3×3 areas of an image contains a constant value (such as all ones), then there are no edges in that area. The result of convolving that area with a mask should be zero. If the numbers in the mask sum to zero, then convolving the mask with a constant area will result in the correct answer of zeros.
- **Second**: the masks should approximate differentiation or amplify the slope of the edge. The simple example [-1 0 1] given earlier showed how to amplify the slope of the edge.

Methods of Detection Edge

The number of masks used for edge detection is almost limitless. Research have used different techniques to derive masks, some of will be illustrated in the following section.

1-Sobel Operator: The Sobel edge detection masks look for edges in both the horizontal and vertical directions and then combine this information into a single metric. The masks are as follows:

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Row Mask

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

Column Mask

Methods of Detection Edge

These masks are each convolved with the image. At each pixel location we now have two numbers: S1, corresponding to the result form the row mask and S2, from the column mask. We use this numbers to compute two matrices, the edge magnitude and the edge direction, which are defined as follows:

$$\text{Edge Magnitude} = \sqrt{S_1{}^2 + S_2{}^2}$$

$$\text{Edge Direction} = \text{Tan -1} \ \left[\frac{S_1}{S_2}\right]$$

Methods of Detection Edge

2-Prewitt Operator: The Prewitt is similar to the Sobel but with different mask coefficients. The masks are defined as follows:

$$\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Row Mask

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

Column Mask

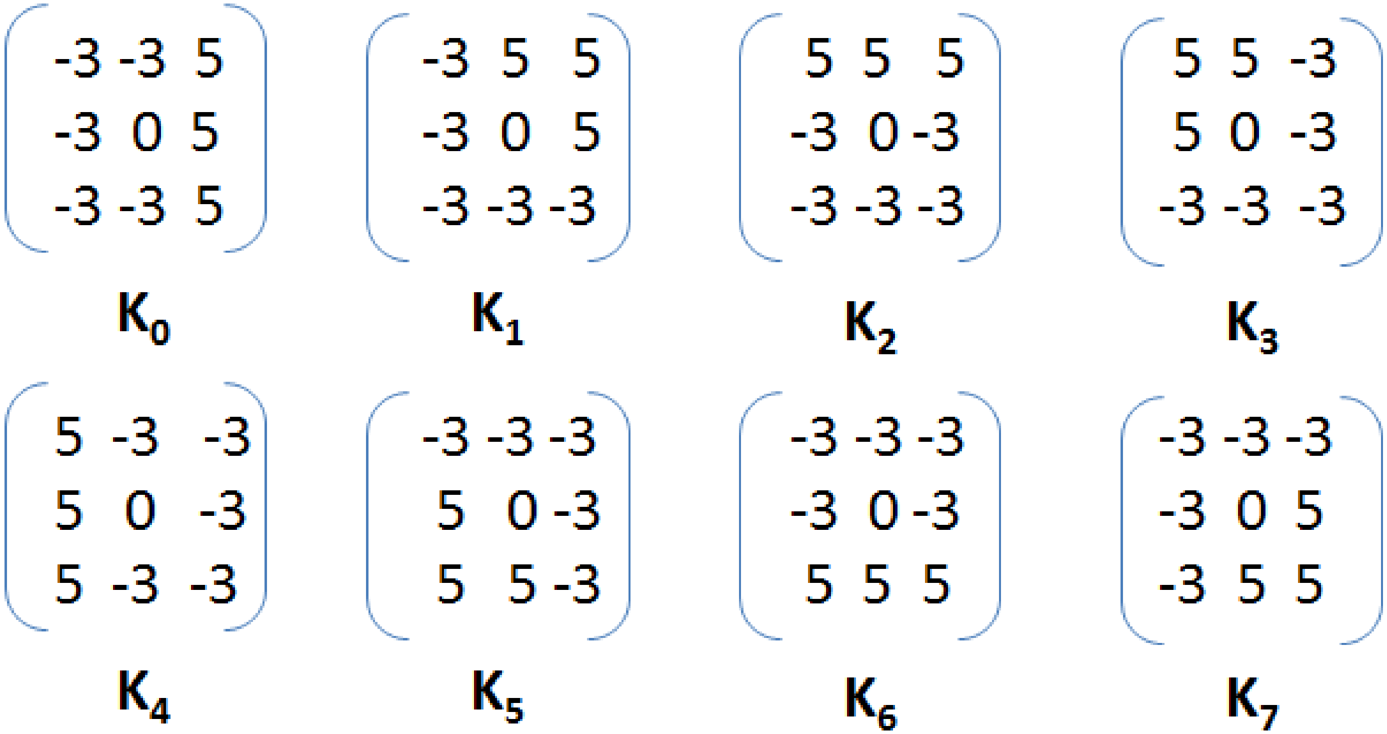
These masks are each convolved with the image. At each pixel location we find two numbers: P1 corresponding to the result from the row mask and P2 from the column mask. We use these results to determine two metrics, the edge magnitude and edge direction, which are defined as follows:

$$\text{Edge Magnitude} = \sqrt{{P_1}^2 + {P_2}^2}$$

$$\text{Edge Direction} = \text{Tan -1} \left[\frac{P_1}{P_2}\right]$$

Methods of Detection Edge

3-Kirsch Compass Mask: is called compass masks because they are define single mask and rotation the masks to the eight major orientations.



$$\begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix} \quad \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix} \quad \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \quad \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

K_0 K_1 K_2 K_3

$$\begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix} \quad \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} \quad \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} \quad \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix}$$

K_4 K_5 K_6 K_7

Methods of Detection Edge

The edge magnitude is defined as the maximum value found by the convolution of each of the mask, with the image.[Given a pixel, there are eight directions you can travel to a neighboring pixel (above, below , left ,right ,upper left, upper right, lower left, lower right). Therefore there are eight possible directions for an edge. The directional edge detectors can detect an edge in only one of the eight directions. If you want to detect only left to right edges, you would use only one of eight masks. If; however you want to detect all of the edges, you would need to perform convolution over an image eight times using each of the eight masks].

Methods of Detection Edge

4-Robinson compass masks: The Robinson compass are used in a manner similar to the kirsch but are easier to implemented because rely only compute in (0,1,2).

5-Laplacian Operators: the Laplacian operator described here are similar to the ones used for pre-processing (as described in enhancement filter). The three Laplacian masks that follow represent different approximation of the Laplacian masks are rationally symmetric, which means edges at all orientation contribute to the result. They are applied by selecting one mask and convolving it with the image selecting one mask and convolving it with the image.

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Methods of Detection Edge

These masks differ from the Laplacian type previously described in that the center coefficients have been decreased by one. So, if we are only interested in edge information, the sum of the coefficients should be zero. If we want to retain most of the information the coefficient should sum to a number greater than zero. Consider an extreme example in which the center coefficient value will depend most heavily up on the current value, with only minimal contribution from surrounding pixel values.

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Methods of Detection Edge

6-Other Edge Detection Methods

Two other methods using Gaussian and homogeneity/difference operators are given below:

$$\begin{pmatrix} 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & -2 & -3 & -3 & -3 & -2 & 0 \\ -1 & -3 & 5 & 5 & 5 & -3 & -1 \\ -1 & -3 & 5 & 16 & 5 & -3 & -1 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & -2 & -3 & -3 & -3 & -2 & 0 \\ -1 & -3 & 5 & 5 & 5 & -3 & -1 \end{pmatrix}$$

Gaussian edge detector has the advantage that the details in the output image can be adjusted by varying the width of the convolution mask. A wider mask eliminates small or fine edges and detects only large, significant edges.

Methods of Detection Edge

Other than by masking, edge detection can also be performed by subtraction. Two methods that use subtraction to detect the edge are Homogeneity operator and Difference operator.

The **homogeneity operator** subtracts each of the pixels next to the center of the n×n area (where n is usually 3) from the center pixel. The result is the maximum of the absolute value of these subtractions. Subtraction in a homogenous region produces zero and indicates an absence of edges. A high maximum of the subtractions indicates an edge. This is a quick operator since it performs only subtraction- eight operations per pixel and no multiplication. This operator then requires thresholding. If there is no thresholding then the resulting image looks like a faded copy of the original. Generally thresholding at 30 to 50 gives good result.

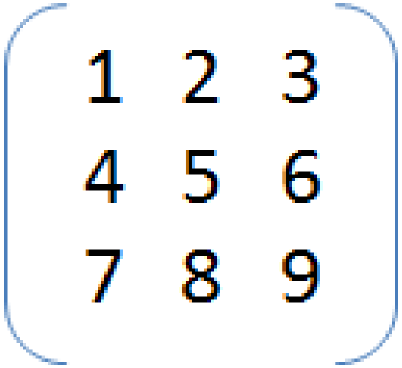
Methods of Detection Edge

The thresholding can be varied depending upon the extent of edge detection desired.

The **difference operator** performs differentiation by calculating the differences between the pixels that surround the center pixel of an n×n area. This operator finds the absolute value of the difference between the opposite pixels, the upper left minus the lower right, upper right minus the lower left, left minus right, and top minus bottom. The result is the maximum absolute value. as in the homogeneity case, this operator requires thresholding. But it is quicker than the homogeneity operator since it uses four integer subtractions as against eight subtractions in homogeneity operator per pixel.

Methods of Detection Edge

Shown below is how the two operators detect the edge:
Consider an image block with center pixel intensity 5,

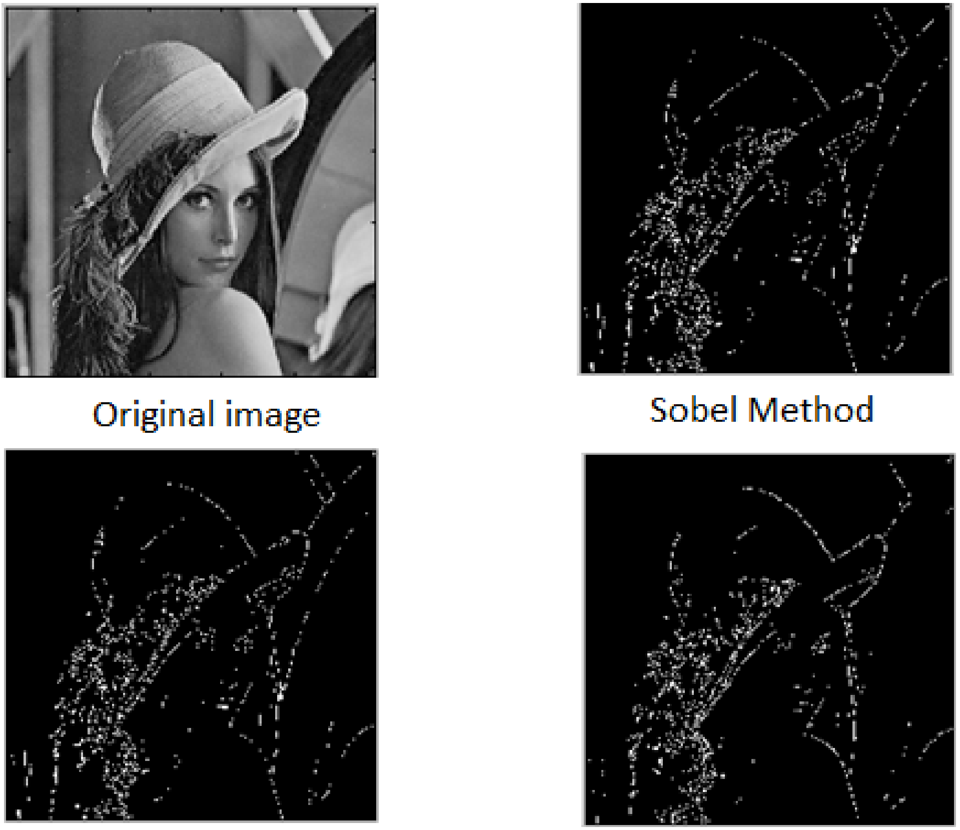


$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Output of **homogeneity operator** is:
Max of {| 5-1 |, | 5-2 |, | 5-3 |, | 5-4 |, | 5-6 |, | 5-7 |, | 5-8 |, | 5-9 | } = 4

Output of **difference operator** is:
Max of {| 1-9 |, | 7-3 |, | 4-6 |, | 2-8 | } = 8

Methods of Detection Edge



Original image

Sobel Method

Prewitt Method

Roberts Method

Example of Edge Operators