

## Image Algebra

There are **two primary categories** of algebraic operations applied to image:

### 1. Arithmetic operations.

Such as: (Addition, subtraction, division and multiplications).

### 2. Logic operations.

Such as:(AND, OR and NOT). These operations which require **only one image**, and are done on a pixel –by-pixel basis.

### Arithmetic operations

To apply the arithmetic operations to two images, we simply operate on corresponding pixel values. For example to add image I1 and I2 to create I3:

$$\begin{array}{c} \mathbf{I_1} \\ \left( \begin{array}{ccc} 3 & 4 & 7 \\ 3 & 4 & 5 \\ 2 & 4 & 6 \end{array} \right) \end{array} + \begin{array}{c} \mathbf{I_2} \\ \left( \begin{array}{ccc} 6 & 6 & 6 \\ 4 & 2 & 6 \\ 3 & 5 & 5 \end{array} \right) = \begin{array}{c} \mathbf{I_3} \\ \left( \begin{array}{ccc} 3+6 & 4+6 & 7+6 \\ 3+4 & 4+2 & 5+6 \\ 2+3 & 4+5 & 6+5 \end{array} \right) = \left( \begin{array}{ccc} 9 & 10 & 13 \\ 7 & 6 & 11 \\ 5 & 9 & 11 \end{array} \right)
 \end{array}$$

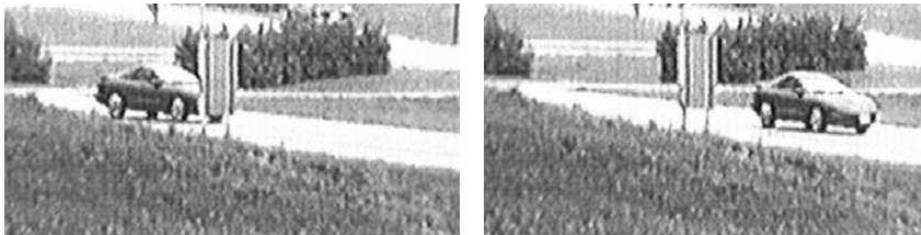
- Addition is used to combine the information in two images. Applications include development of image restoration algorithm for molding additive noise, and special effects, such as image morphing in motion pictures.
- Subtraction of two images is often used to detect motion consider the case where nothing has changed in a sense; the image resulting from subtraction of two sequential image is filled with zero-a black image.If something has moved in the scene, subtraction produces a nonzero result at the location of movement. Applications include Object tracking , Medical imaging, Law enforcement and Military applications.

- Multiplication and Division are used to adjust the brightness of an image. One image typically consists of a constant number greater than one. Division of the pixel values by a number greater than one will darken the image (Brightness adjustment is often used as a processing step in image enhancement).



a. First Original image    b. Second Original    c. Addition of two images

**Figure (2.3): Image Addition.**



a Original scene

b Same scene later



c. Subtraction of scene a from scene b



a. Cameraman image



b. X-ray image of hand



c. Multiplication of two images

**Figure (2.5): Image Multiplication.**

a. Original image



b. Image divided by value &lt; 1



c. Image divided by value &gt; 1

**Figure (2.6): Image Division.**

### Logic operations

The logic operations AND, OR and NOT form a complete set, meaning that any other logic operation (XOR, NOR, NAND) can be created by a combination of these basic elements. They operate in a bit-wise fashion on pixel data.

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

**AND**

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

**NAND**

The logic operation AND and OR are used to combine the information in two images. They may be done for special effects, but a more useful application for image analysis is to perform a **masking operation**. Use **AND and OR as a simple method to extract a Region of Interest from an image**, if more sophisticated graphical methods are not available. **Example:** A white square ANDed with an image will allow only the portion of the image coincident with the square to appear in the output image with the background turned black; and a black square ORd with an image will allow only the part of the image corresponding to the black square to appear in the output image but will turn the rest of the image white. This process is called image masking.

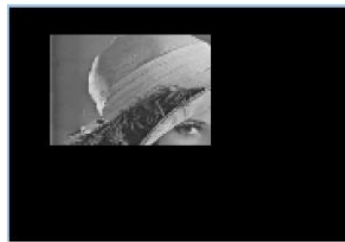


Original image

AND



Anding Mask



Output image

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

**OR**

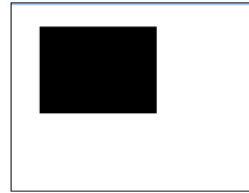
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

**NOR**

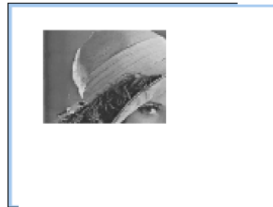


Original image

OR



OR mask



Output image

The NOT operation creates a negative of the original image, by inverting each bit within each pixel value.

A	Q
0	1
1	0

**NOT**

In 256-gray level images the not operation can be done by the following equation

$$Q(i, j) = 255 - P(i, j)$$



Original image



Output image after NOT

The other operations of interest for ROI image **geometric operation** such as translation and rotation.

### **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 binary (two level) image and is often used as preprocessing step in the extraction of object features, such as shape, area, or perimeter.

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.

**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 1111100.

this is equivalent to dividing by eight( 8 ), 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:

$$256 \div 8 = 32$$

The general case requires us to mask k bits, where  $2^k$  is divided into the original gray-level range to get the quantized range desired. Using this method, we can reduce the number of gray levels to any power of 2:

2,4,6,8, 16, 32, 64 or 128.

□ Image quantization by masking to **128** gray level, 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.

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



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

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.

## 2- Quantization of Spatial Coordinates Methods

In this methods, the size of image will decrease by tacking the pixel and it's neighbors and convert it to one pixel. there are three methods for quantization spatial coordinates

- a) Average method: in this method we take the average of the gray levels pixels with determine mask size on the original image.
- b) Median: in this method the pixels in mask will sort in descent and take the pixel in the middle.

c) Decimation or Down sampling: in this method, the pixels will neglected by factor  $k$  (for example if  $k=2$ ) we take every row and column and neglect the other.