

Lecture 3

Fourth stage

Medical Physical Department



Medical Imaging Processing

Medical Image Algebra Operation

By

Asst. Prof. Dr. Mehdi Ebady Manaa

Medical Imaging Processing (Image Types and Formats)

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.

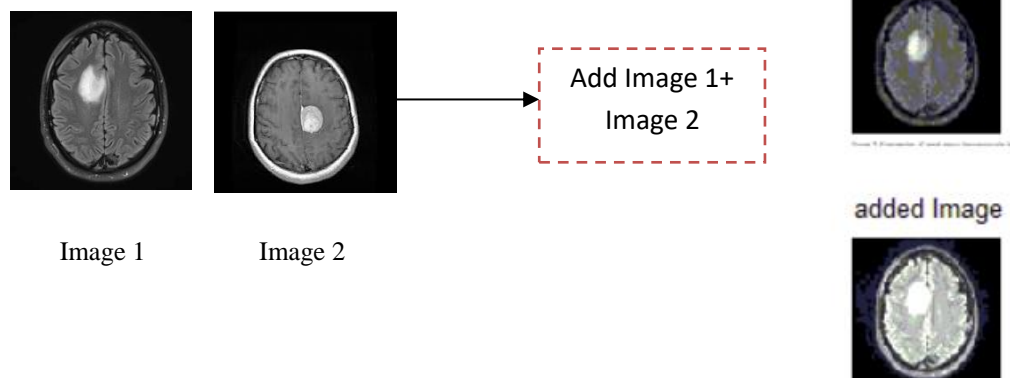
Such as:(AND, OR and NOT). These operations which require **only one image**, and are done on a pixel –by-pixel basis.

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:

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

- Addition is used to combine the information in two images. Applications include development of image restoration algorithm for molding additive noise. See this example for two images.



```

x=imread('gray(1).png');
y=imresize(x,[100,100]);
subplot(3,1,1), imshow(y), title('Original Image');
x1=imread('gray(1).png');
y1=imresize(x1,[100,100]);
subplot(3,1,2), imshow(y1), title('second Image');
xx=imadd(y,y1);
subplot(3,1,3); imshow(xx); title('added Image');

```

- 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 the image is filled with zero-a black image.

Original Image



- The command for subtraction is **subtract(x,y)**

```

x=imread('gray(1).png');
y=imresize(x,[100,100]);
subplot(3,1,1), imshow(y),
title('Original Image');
x1=imread('gray(1).png');
y1=imresize(x1,[100,100]);
subplot(3,1,2), imshow(y1),
title('second Image');
xx=imsubtract(y,y1);
subplot(3,1,3); imshow(xx);

```

second Image



added Image



- Multiplication and Division are used to adjust the brightness of an image. One image typically consists of a constant number greater than one. Division (**imdivide**) 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).



Image Multiply
with itself

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**,

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. 11000000 and $10000000=10000000$

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 11110000. 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:

:

$$256 \div 8 = 32$$

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.