Image Algebra (LEC 6)

6-1 introduction:

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

1. Arithmetic operations.

2. Logic operations.

Addition, subtraction, division and multiplications comprise the arithmetic operations, while AND, OR and NOT makeup the logic operations. These operations which require only one image, and are done on a pixel –by-pixel basis.

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:

$$\underset{I_1}{\begin{pmatrix} 3 & 4 & 7 \\ 3 & 4 & 5 \\ 2 & 4 & 6 \end{pmatrix}} + \underset{I_2}{\begin{pmatrix} 6 & 6 & 6 \\ 4 & 2 & 6 \\ 3 & 5 & 5 \end{pmatrix}} = \underset{I_3}{\begin{pmatrix} 3+6 & 4+6 & 7+6 \\ 3+4 & 4+2 & 5+6 \\ 2+3 & 4+5 & 6+5 \end{pmatrix}} = \begin{pmatrix} 9 & 10 & 13 \\ 7 & 6 & 11 \\ 5 & 9 & 11 \end{pmatrix}$$

• 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. Multiplication of the pixel values by a number grater than one will darken the image (Brightness adjustment is often used as a processing step in image enhancement).

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.

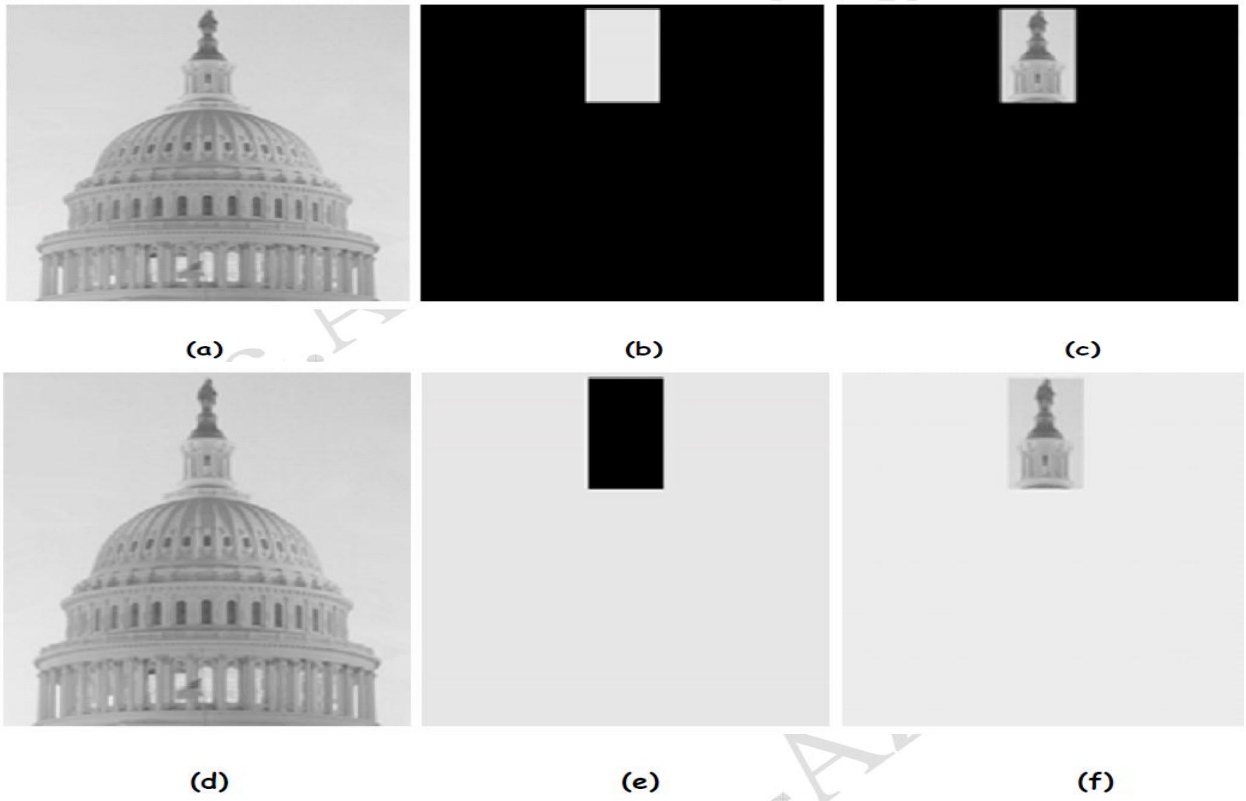


FIGURE (6-1). (a) Original image. (b)AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

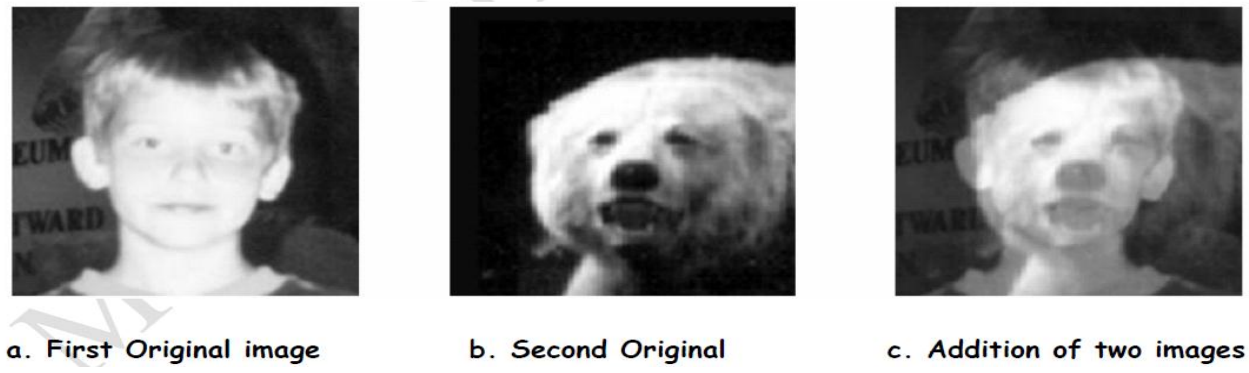


Figure (6.2): Image Addition

Figure (6.3): Image Subtraction

a. Cameraman image b. X-ray image of hand c. Multiplication of two images

Figure (6.4): Image Multiplication

a. Original image b. Image divided by value<1 c. Image divided by value >1

Figure (6.5): Image Division

Example: A logic AND is performed on two images, suppose the two corresponding pixel values are (111)10 is one image and (88)10 in the second image. The corresponding bit strings are:

$$(111)_{10} \longrightarrow 01101111_2$$

$$\text{AND}$$

$$(88)_{10} \longrightarrow 01011000_2$$

$$01001000$$

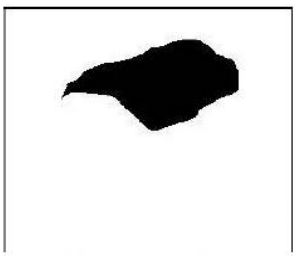
The logic operation **AND** and **OR** are used to combine the information in two images. They may be done for special effects, but a more useful application for image analysis is to perform a masking operation. Use **AND** and **OR** as simple method to extract a Region of Interest from an image, if 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. The NOT operation creates a negative of the original image, by inverting each bit within each pixel value.

a. Original image

b. Image mask (AND)

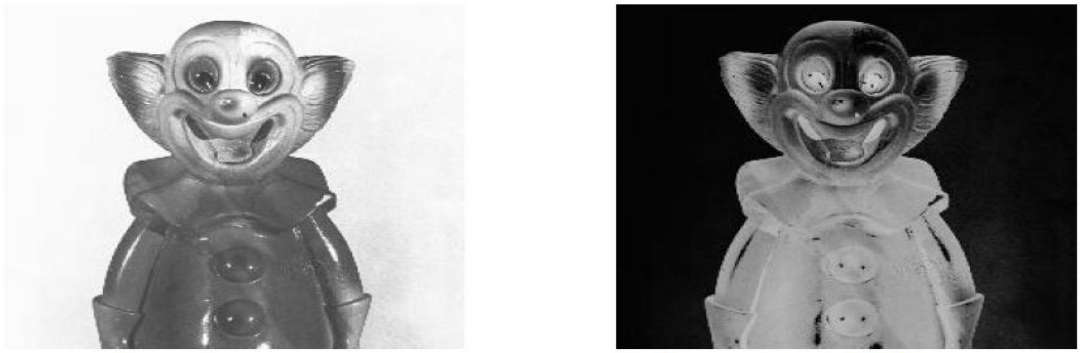
c. ANDing a and b



d. Image mask (OR)

e. ORing a and d

Figure (6.6): Image masking



a. Original image

b. Image after NOT operation.

Figure (6.7): Complement Image