IE404

Digital Image Processing

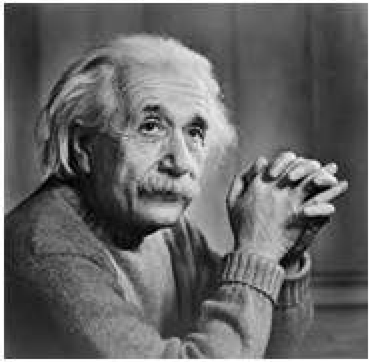
Instructor: Manish Khare

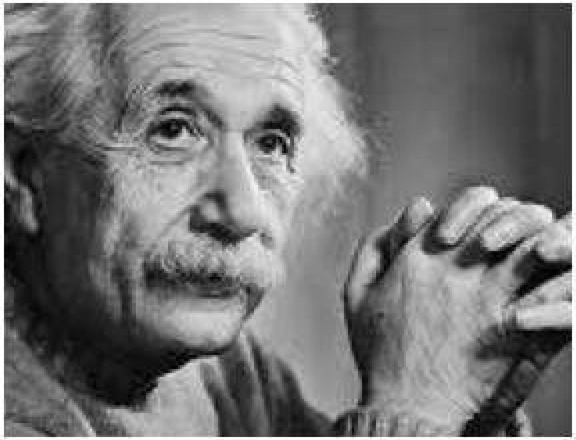
Lecture – 6-8

Zooming

- Zooming simply means enlarging a picture in a sense that the details in the image became more visible and clear.

- Zooming an image has many wide applications ranging from zooming through a camera lens, to zoom an image on internet e.t.c.





Optical Zoom vs digital Zoom

- Optical Zoom:
  - The optical zoom is achieved using the movement of the lens of your camera.
  - An optical zoom is actually a true zoom. The result of the optical zoom is far better then that of digital zoom.
  - In optical zoom, an image is magnified by the lens in such a way that the objects in the image appear to be closer to the camera.
  - In optical zoom the lens is physically extend to zoom or magnify an object.

- Digital Zoom:
  - Digital zoom is basically image processing within a camera. During a digital zoom, the center of the image is magnified and the edges of the picture got crop out.
  - Due to magnified center, it looks like that the object is closer to you.
  - During a digital zoom, the pixels got expand , due to which the quality of the image is compromised.
  - The same effect of digital zoom can be seen after the image is taken through your computer by using an image processing toolbox / software, such as Photoshop.

Zooming methods

- Although there are many methods that does this job, but we are going to discuss the most common of them here.

  - Pixel replication or (Nearest neighbor interpolation)
  - Zero order hold method
  - Zooming K times

Method 1: Pixel replication

- It is also known as Nearest neighbor interpolation.
- As its name suggest, in this method, we just replicate the neighboring pixels.
- Zooming is nothing but increase amount of sample or pixels.

Working of Pixel Replication

- In this method we create new pixels form the already given pixels.
- Each pixel is replicated in this method n times row wise and column wise and you got a zoomed image. Its as simple as that.

For example:

- if you have an image of 2 rows and 2 columns and you want to zoom it twice or 2 times using pixel replication, here how it can be done.

For a better understanding, the image has been taken in the form of matrix with the pixel values of the image.

1	2
3	4

The above image has two rows and two columns, we will first zoom it row wise after that zoom updated matrix column wise.

Row wise zooming:

- When we zoom it row wise, we will just simple copy the rows pixels to its adjacent new cell.

- Here how it would be done.

1	1	2	2
3	3	4	4

As you can see that in the above matrix, each pixel is replicated twice in the rows

Column size zooming:

- The next step is to replicate each of the pixel column wise, that we will simply copy the column pixel to its adjacent new column or simply below it.
- Here how it would be done

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

New image size:

- As it can be seen from the above example, that an original image of 2 rows and 2 columns has been converted into 4 rows and 4 columns after zooming. That means the new image has a dimensions of

- (Original image rows * zooming factor, Original Image cols * zooming factor)

Advantage and disadvantage

- One of the advantage of this zooming technique is, it is very simple. You just have to copy the pixels and nothing else.

- The disadvantage of this technique is that image got zoomed but the output is very blurry. And as the zooming factor increased, the image got more and more blurred. That would eventually result in fully blurred image.

Method 2: Zero order hold

- Zero order hold method is another method of zooming. It is also known as zoom twice.

- Because it can only zoom twice. We will see in the below example that why it does that

Working of Zero order hold

- In zero order hold method, we pick two adjacent elements from the rows respectively and then we add them and divide the result by two, and place their result in between those two elements.

- We first do this row wise and then we do this column wise.

- Lets take an image of the dimensions of 2 rows and 2 columns and zoom it twice using zero order hold.

1	2
3	4

Row Wise Zooming

- First we will zoom it row wise and then column wise

1	1	2
3	3	4

As we take the first two numbers : (2 + 1) = 3 and then we divide it by 2, we get 1.5 which is approximated to 1. The same method is applied in the row 2

Column wise zooming

1	1	2
2	2	3
3	3	4

We take two adjacent column pixel values which are 1 and 3. We add them and got 4. 4 is then divided by 2 and we get 2 which is placed in between them. The same method is applied in all the columns

- As you can see that the dimensions of the new image are 3 x 3 where the original image dimensions are 2 x 2.

- So it means that the dimensions of the new image are based on the following formula

(2(number of rows) minus 1) X (2(number of columns) minus 1)

Advantages and disadvantage

- One of the advantage of this zooming technique , that it does not create as blurry picture as compare to the nearest neighbour interpolation (pixel replication) method.

- But it also has a disadvantage that it can only run on the power of 2

Reason behind twice zooming

- Consider the above image of 2 rows and 2 columns. If we have to zoom it 6 times, using zero order hold method , we can not do it. As the formula shows us this.

- It could only zoom in the power of 2 2,4,8,16,32 and so on.

- Even if you try to zoom it, you can not. Because at first when you will zoom it two times, and the result would be same as shown in the column wise zooming with dimensions equal to 3x3. Then you will zoom it again and you will get dimensions equal to 5 x 5. Now if you will do it again, you will get dimensions equal to 9 x 9.

- Whereas according to the formula of yours the answer should be 11x11. As (6(2) minus 1) X (6(2) minus 1) gives 11 x 11.

Method 3: K-Times Zooming

- K times is the third zooming method we are going to discuss. It is one of the most perfect zooming algorithm discussed so far.

- It caters the challenges of both twice zooming and pixel replication.

- K in this zooming algorithm stands for zooming factor

Working of K-Times Zooming

- First of all, you have to take two adjacent pixels as you did in the zooming twice. Then you have to subtract the smaller from the greater one. We call this output (OP).

- Divide the output(OP) with the zooming factor(K). Now you have to add the result to the smaller value and put the result in between those two values.

- Add the value OP again to the value you just put and place it again next to the previous putted value. You have to do it till you place k-1 values in it.

- Repeat the same step for all the rows and the columns , and you get a zoomed images.

Example

- Suppose you have an image of 2 rows and 3 columns, which is given below. And you have to zoom it three times.

15	30	15
30	15	30

- K in this case is 3. K = 3.
- The number of values that should be inserted is k-1 = 3-1 = 2.

Row Wise Zooming

- Take the first two adjacent pixels. Which are 15 and 30.
- Subtract 15 from 30. 30-15 = 15.
- Divide 15 by k. 15/k = 15/3 = 5. We call it OP.(where op is just a name)
- Add OP to lower number. 15 + OP = 15 + 5 = 20.
- Add OP to 20 again. 20 + OP = 20 + 5 = 25.
- We do that 2 times because we have to insert k-1 values.
- Now repeat this step for the next two adjacent pixels. It is shown in the first table.
- After inserting the values, you have to sort the inserted values in ascending order, so there remains a symmetry between them. It is shown in the second table

Table 1.

15	20	25	30	20	25	15
30	20	25	15	20	25	30

Table 2.

15	20	25	30	**25**	**20**	15
30	**25**	**20**	15	20	25	30

Column wise zooming

- The same procedure has to be performed column wise. The procedure include taking the two adjacent pixel values, and then subtracting the smaller from the bigger one. Then after that, you have to divide it by k. Store the result as OP. Add OP to smaller one, and then again add OP to the value that comes in first addition of OP. Insert the new values.

- Here what you got after all that

15	20	25	30	25	20	15
20	21	21	25	21	21	20
25	22	22	20	22	22	25
30	25	20	15	20	25	30

New image size

- The best way to calculate the formula for the dimensions of a new image is to compare the dimensions of the original image and the final image. The dimensions of the original image were 2 X 3. And the dimensions of the new image are 4 x 7.

- The formula thus is:

(K (number of rows minus 1) + 1) X (K (number of cols minus 1) + 1)

Advantages and disadvantages

- The one of the clear advantage that k time zooming algorithm has that it is able to compute zoom of any factor which was the power of pixel replication algorithm , also it gives improved result (less blurry) which was the power of zero order hold method. So hence It comprises the power of the two algorithms.

- The only difficulty this algorithm has that it has to be sort in the end, which is an additional step, and thus increases the cost of computation.

Spatial Resolution

- resolution can be defined in many ways.
  - One type of it which is pixel resolution that has been discussed with concept of aspect ratio.

- Here we will discuss another type of resolution which is spatial resolution.

- Spatial resolution states that the clarity of an image that cannot be determined by the pixel resolution.

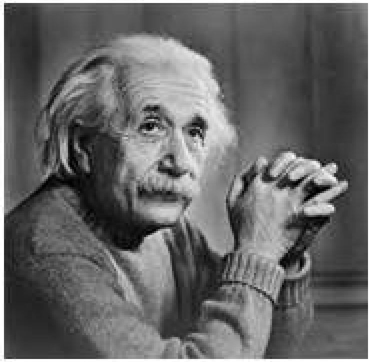
- Spatial resolution can be defined as the smallest discernible detail in an image.

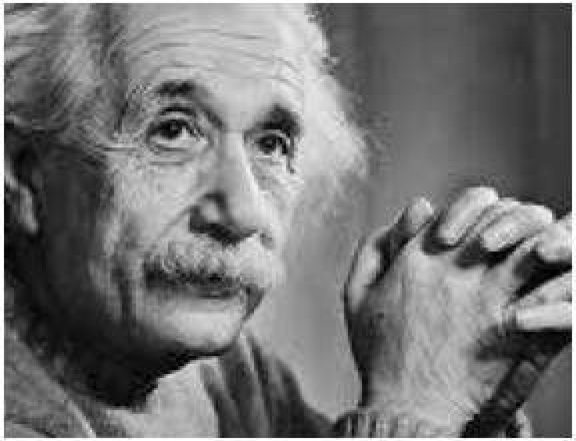
- Or in other way we can define spatial resolution as the number of independent pixels values per inch.

- In short what spatial resolution refers to is that we cannot compare two different types of images to see that which one is clear or which one is not.

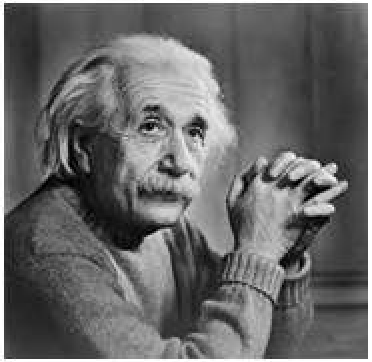
- If we have to compare the two images, to see which one is more clear or which has more spatial resolution, we have to compare two images of the same size.

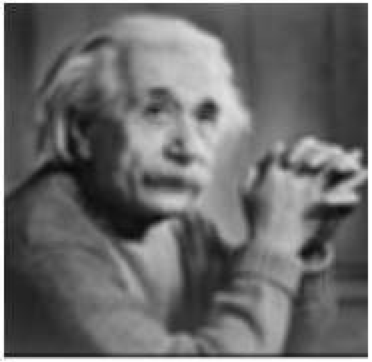
You cannot compare these two images to see the clarity of the image





So in order to measure spatial resolution , the pictures below would server the purpose.





Now you can compare these two pictures. Both the pictures has same dimensions which are of 227 X 222. Now when you compare them, you will see that the picture on the left side has more spatial resolution or it is more clear then the picture on the right side. That is because the picture on the right is a blurred image.

Measuring spatial resolution

- Since the spatial resolution refers to clarity, so for different devices, different measure has been made to measure it.
- For example
  - Dots per inch
  - Lines per inch
  - Pixels per inch

- Dots per inch
  - Dots per inch or DPI is usually used in monitors.
- Lines per inch
  - Lines per inch or LPI is usually used in laser printers.
- Pixel per inch
  - Pixel per inch or PPI is measure for different devices such as tablets , Mobile phones etc.

Pixels Per Inch

- Pixel density or Pixels per inch is a measure of spatial resolution for different devices that includes tablets, mobile phones.

- The higher is the PPI, the higher is the quality. In order to more understand it, that how it calculated. Lets calculate the PPI of a mobile phone.

- The Samsung galaxy S9/S10 has PPI or pixel density of 570. But how does it is calculated?

First of all we will Pythagoras theorem to calculate the diagonal resolution in pixels.

It can be given as: $c = \sqrt{a^2 + b^2}$

Where a and b are the height and width resolutions in pixel and c is the diagonal resolution in pixels.

For Samsung galaxy S9/S10, it is 2960 X 1440 pixels.

So putting those values in the equation gives the result

C = 3305.6865

Now we will calculate PPI

PPI = c / diagonal size in inches

The diagonal size in inches of Samsung galaxy s4 is 5.8 inches , which can be confirmed from anywhere.

PPI = 3305.6865 /5.8

PPI = 569.94

PPI = 570 (approx)

That means that the pixel density of Samsung galaxy s9/s10 is 570 PPI.

Dots per inch

- The dpi is often relate to PPI, whereas there is a difference between these two. DPI or dots per inch is a measure of spatial resolution of printers. In case of printers, dpi means that how many dots of ink are printed per inch when an image get printed out from the printer.

- Remember, it is not necessary that each Pixel per inch is printed by one dot per inch. There may be many dots per inch used for printing one pixel. The reason behind this that most of the color printers uses CMYK model. The colors are limited. Printer has to choose from these colors to make the color of the pixel whereas within pc, you have hundreds of thousands of colors.

- The higher is the dpi of the printer, the higher is the quality of the printed document or image on paper.

- Usually some of the laser printers have dpi of 300 and some have 600 or more.

Lines per inch

- When dpi refers to dots per inch, liner per inch refers to lines of dots per inch. The resolution of halftone screen is measured in lines per inch.
- The following table shows some of the lines per inch capacity of the printers.

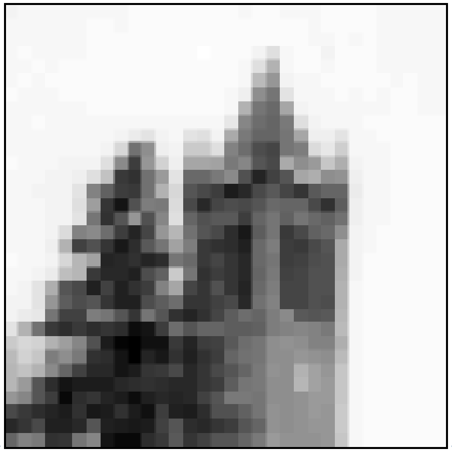
Printer	LPI
Screen printing	45-65 lpi
Laser printer (300 dpi)	65 lpi
Laser printer (600 dpi)	85-105 lpi
Offset Press (newsprint paper)	85 lpi
Offset Press (coated paper)	85-185 lpi

Effect of Spatial Resolution

256x256 pixels

128x128 pixels

64x64 pixels



32x32 pixels

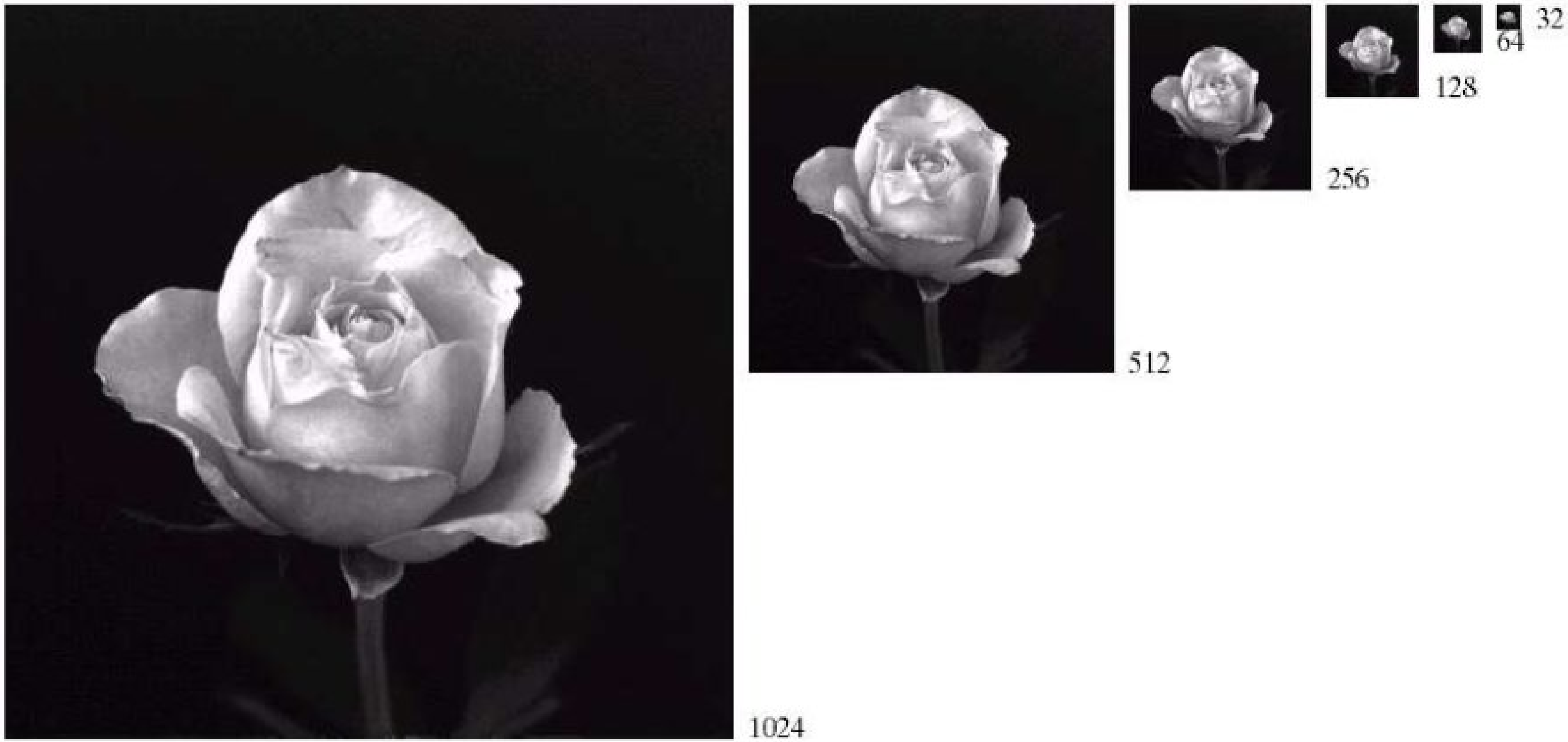
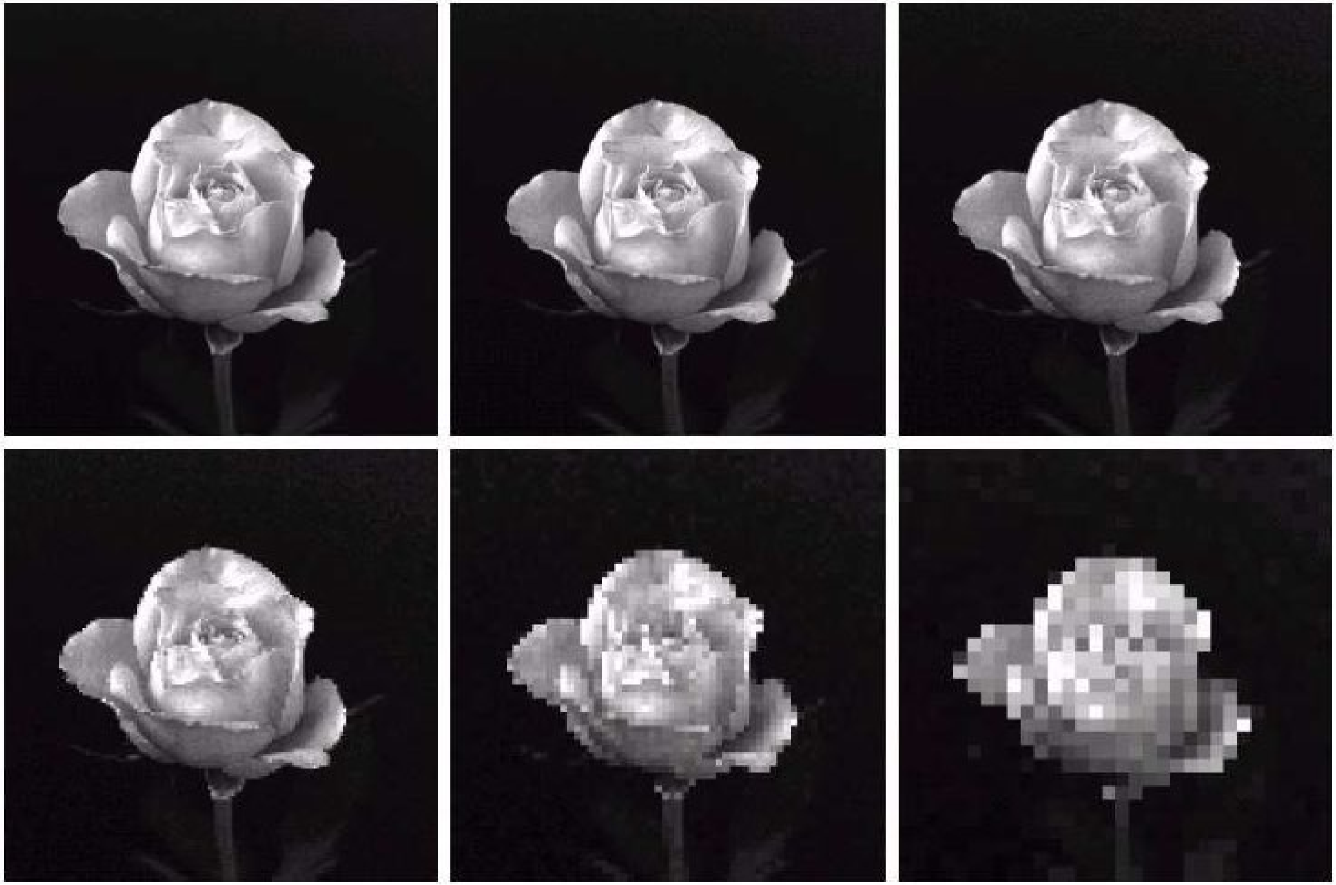


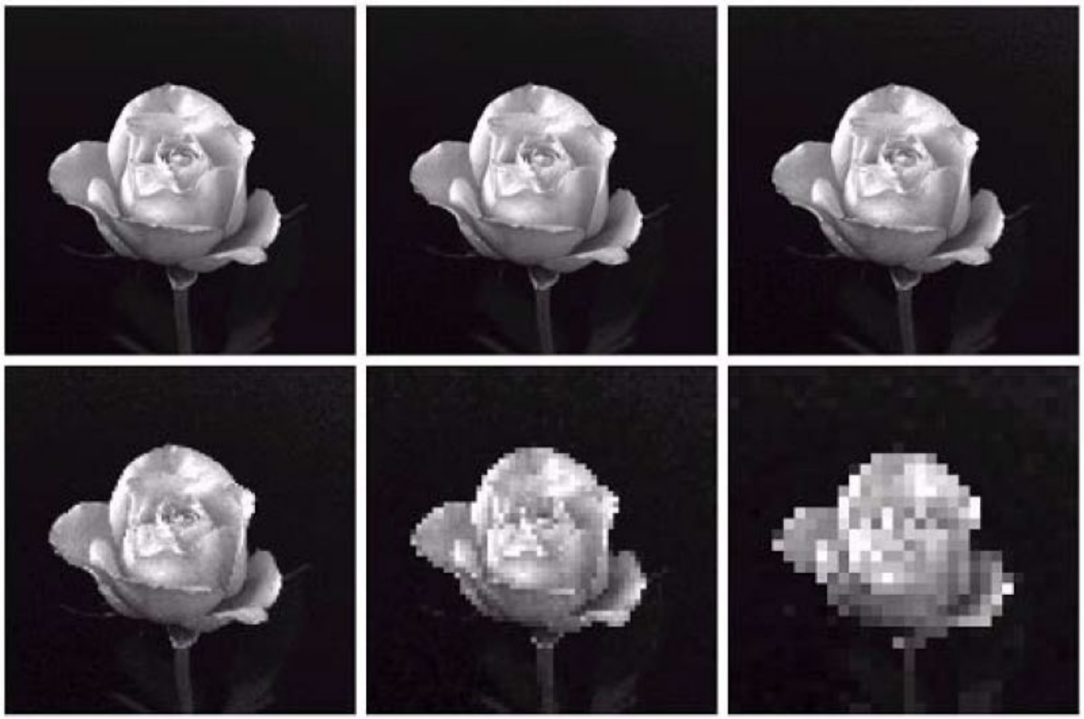
FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.



a b c
d e f

FIGURE 2.20 (a) 1024×1024, 8-bit image. (b) 512×512 image resampled into 1024×1024 pixels by row and column duplication. (c) through (f) 256×256, 128×128, 64×64, and 32×32 images resampled into 1024×1024 pixels.

Checkerboard Effect



a	b	c
d	e	f

(a) 1024x1024

(b) 512x512

(c) 256x256

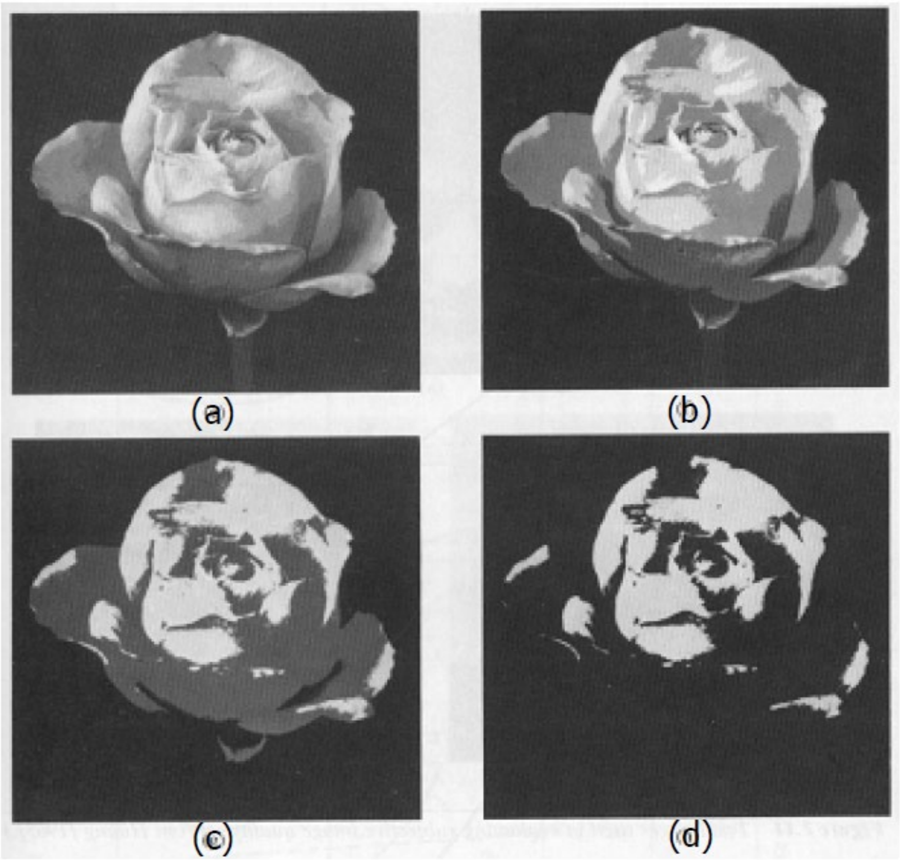
(d) 128x128

(e) 64x64

(f) 32x32

If the resolution is decreased too much, the checkerboard effect can occur.

False Contouring



(a) Gray level = 16

(b) Gray level = 8

(c) Gray level = 4

(d) Gray level = 2

If the gray scale is not enough, the smooth area will be affected.

False contouring can occur on the smooth area which has fine gray scales.

Intensity Resolution

- It refers to the smallest discernible change in intensity level
- Number of intensity levels usually is an integer power of two
- Also refers to Number of bits used to quantize intensity as the intensity resolution
- Which intensity resolution is good for human perception 8 bit, 16 bit, or 32 bit

Difference between Spatial and Intensity Resolution

- Spatial resolution and Resolution intensity are terms used in **image resolution or clarity of image**. In simple terms, images are referred to as blurred or sharp, depending on the intensity of resolution.

- **Intensity of resolution** means the number of pixels per square inch, which determines the clarity or sharpness of an image.

- **Spatial resolution** refers to the number of pixels used in making an image. Images with a higher number of pixels per square inch are sharp and hence said to have a higher Spatial resolution. Such images are very clear.

Effect of Intensity Resolution


256 levels


128 levels

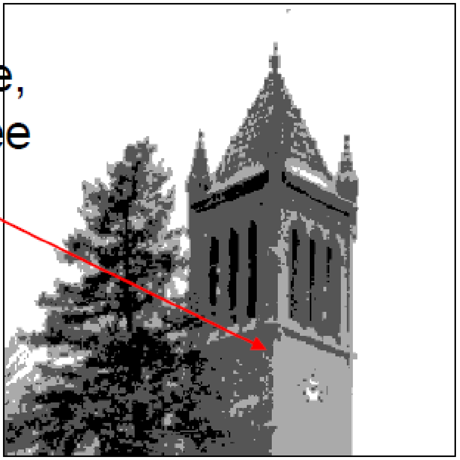

64 levels


32 levels

16 levels

8 levels

In this image, it is easy to see false contour.



4 levels

2 levels

How to select the suitable size and pixel depth of images

- The word “suitable” is subjective: depending on “subject”.

Low detail image

Lena image

Medium detail image

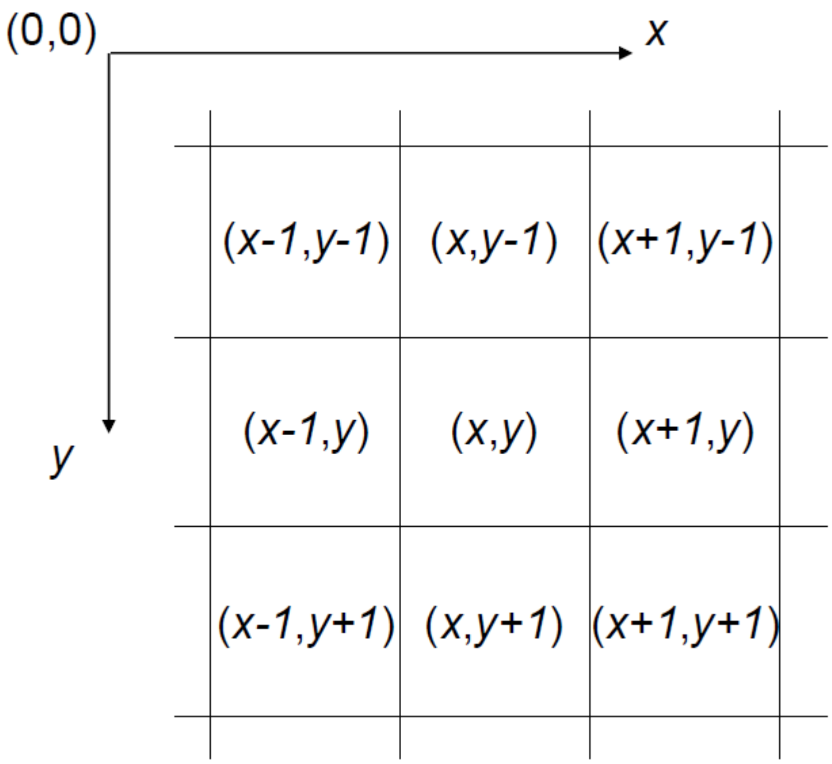
Cameraman image

High detail image

- To satisfy human mind
  - 1. For images of the same size, the low detail image may need more pixel depth.
  - 2. As an image size increase, fewer gray levels may be needed

Basic Relationship between Pixels

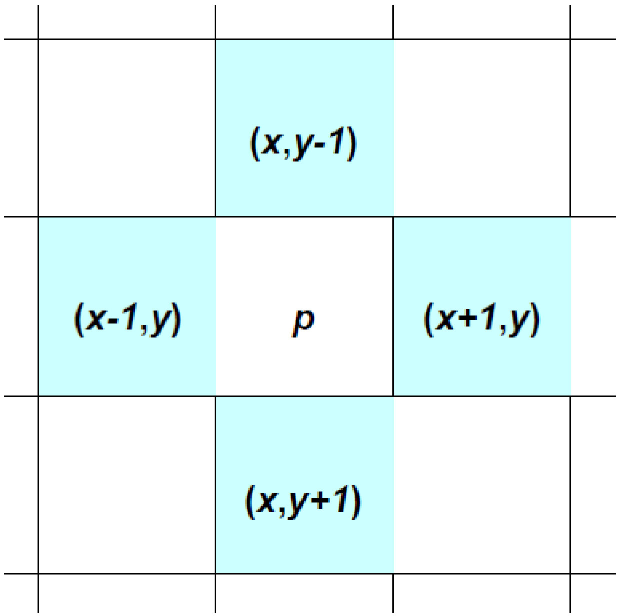
- Neighbors of a pixel
- Connectivity
- Labeling of Connected Components
- Relations, Equivalences, and Transitive Closure
- Distance Measures
- Arithmetic/Logic Operations



Conventional indexing method

Neighbors of a pixel

Neighborhood relation is used to tell adjacent pixels. It is useful for analyzing regions.



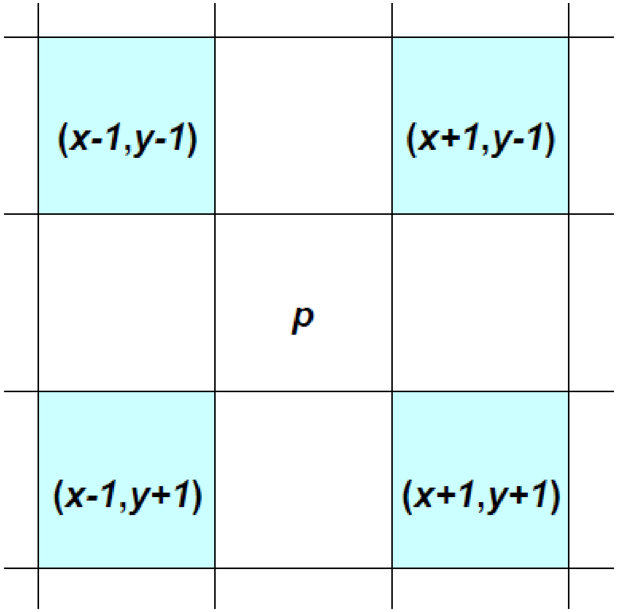
4-neighbors of *p*:

$$N_4(p) = \left\{ \begin{array}{c} (x-1,y) \\ (x+1,y) \\ (x,y-1) \\ (x,y+1) \end{array} \right\}$$

4-neighborhood relation considers only vertical and horizontal neighbors.

Note: $q \in N_4(p)$ implies $p \in N_4(q)$

Neighbors of a pixel



Diagonal neighbors of *p*:

$$N_D(p) = \left\{ \begin{array}{c} (x-1,y-1) \\ (x+1,y-1) \\ (x-1,y+1) \\ (x+1,y+1) \end{array} \right\}$$

Diagonal -neighborhood relation considers only diagonal neighbor pixels.

Neighbors of a pixel

(x-1,y-1)	*(x,y-1)*	*(x+1,y-1)*
(x-1,y)	*p*	*(x+1,y)*
(x-1,y+1)	*(x,y+1)*	*(x+1,y+1)*

8-neighbors of *p*:

$$N_8(p) = \left\{ \begin{array}{c} (x-1,y-1) \\ (x,y-1) \\ (x+1,y-1) \\ (x-1,y) \\ (x+1,y) \\ (x-1,y+1) \\ (x,y+1) \\ (x+1,y+1) \end{array} \right\}$$

8-neighborhood relation considers all neighbor pixels.

Connectivity

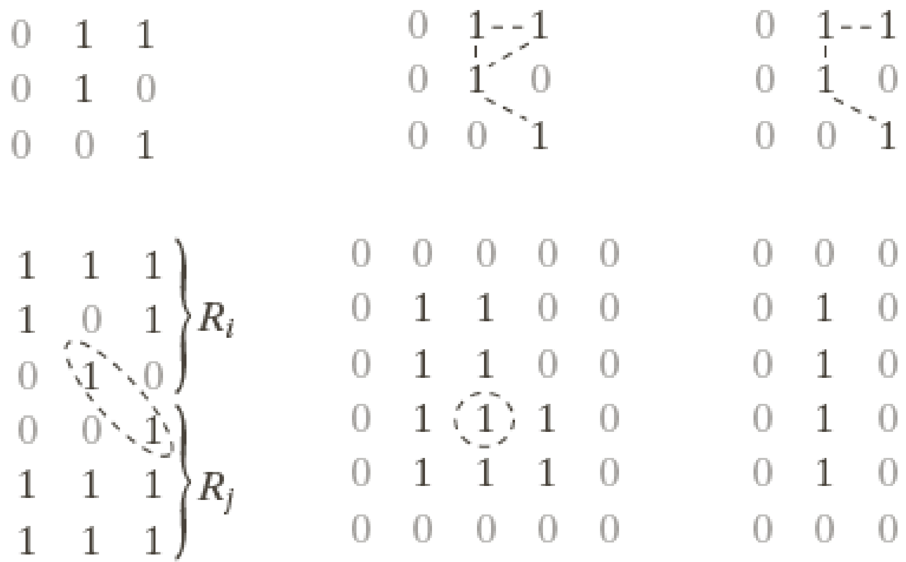
- Let V be the set of gray-level values used to defined connectivity. In a binary image, V={1} if we are referring to connectivity of pixels with value 1. In a grayscale image, the idea is same, but set V typically contains more elements.

- For ex. If we are dealing with the connectivity of pixels whose values are in the range of 0 to 255, set V could be any subset of these 256 values.

- So three types of connectivity:
  - 4- connectivity
  - 8- connectivity
  - m- connectivity

Connectivity

- 4- connectivity :
  - 2 pixels p and q with values from V are 4-connected if q is in the set N4(p).
- 8-adjacency connectivity :
  - 2 pixels p and q with values from V are 8-connected if q is in the set N8(p)
- m- connectivity (mixed connectivity):
  - 2 pixels p and q with values from V are m-connected if
    - q is in the set N4(p) or
    - q is in the set ND(p) and the set $N4(p) \cap N4(q)$ is empty.
    - (the set of pixels that are 4-neighbors of both p and q whose values are from V)

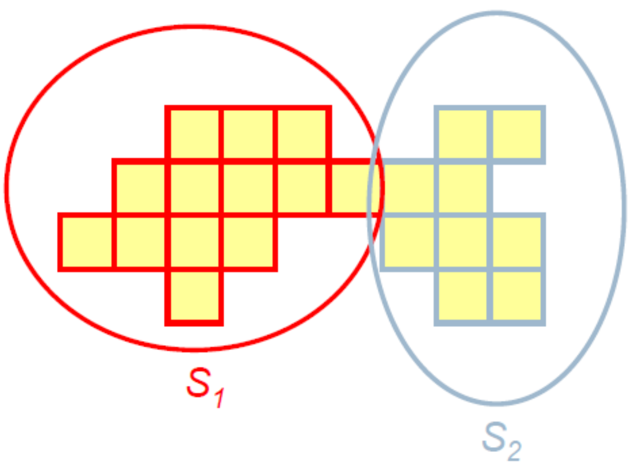


a b c
d e f

FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) m-adjacency. (d) Two regions that are adjacent if 8-adjecency is used. (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

Adjacency

- A pixel *p* is *adjacent* to pixel *q* is they are connected.
  - Two image subsets *S1* and *S2* are adjacent if some pixel in *S1* is adjacent to some pixel in *S2*.
- We can define type of adjacency: 4-adjacency, 8-adjacency or m-adjacency depending on type of connectivity



Path

- A ***path*** from pixel p at (x,y) to pixel q at (s,t) is a sequence of distinct pixels:

  $(x0,y0), (x1,y1), (x2,y2),\ldots, (xn,yn)$

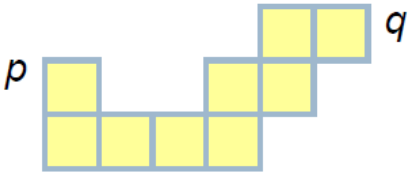
- such that

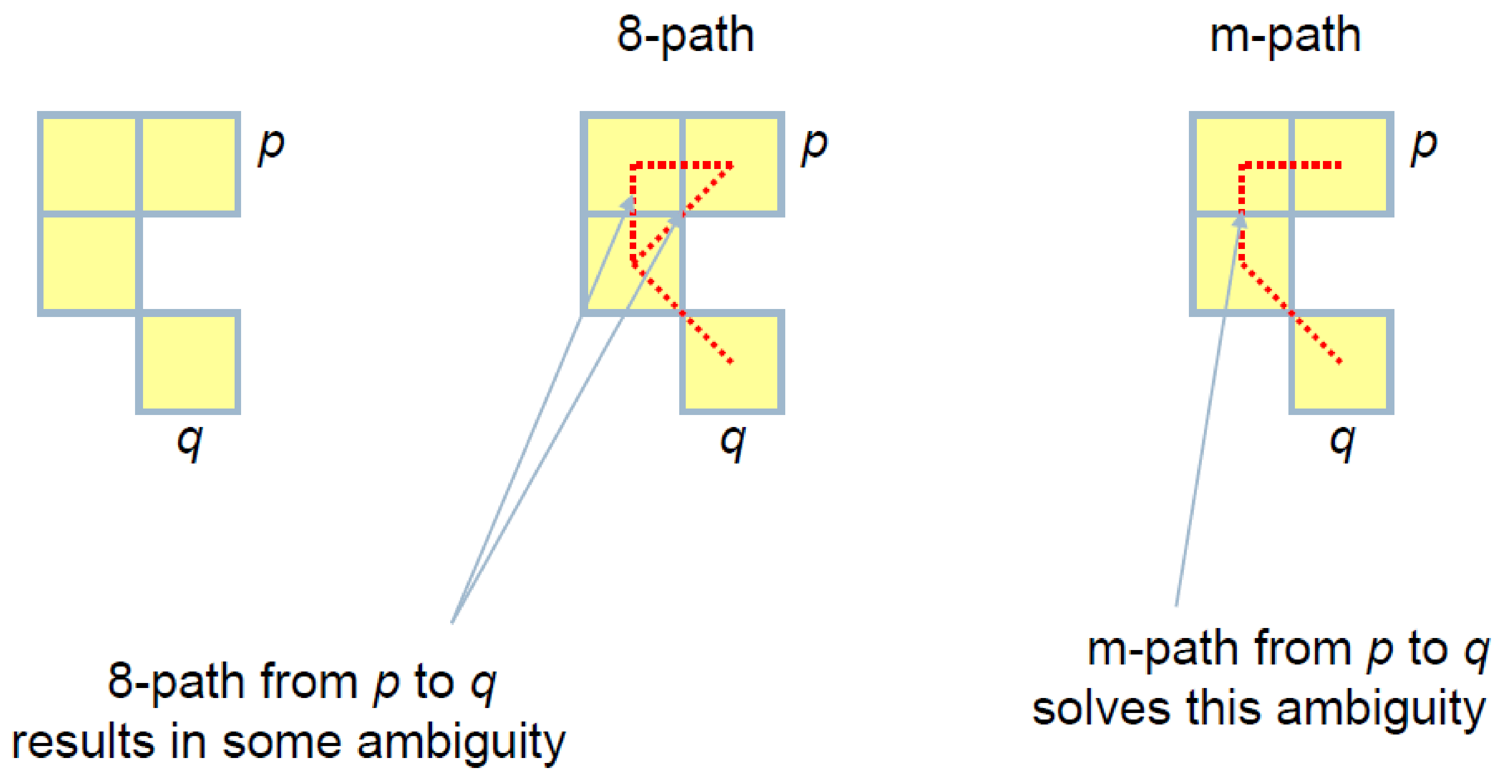
  $(x0,y0) = (x,y)$ and $(xn,yn) = (s,t)$

- and

  (xi,yi) is adjacent to $(xi\text{-}1,yi\text{-}1)$, $i = 1,\ldots,n$

- We can define type of path: 4-path, 8-path or m-path depending on type of adjacency.




8-path
m-path
p
q
p
q
p
q
8-path from p to q
results in some ambiguity
m-path from p to q
solves this ambiguity

Distance

- For pixel p, q, and z with coordinates (x,y), (s,t) and (u,v),

D is a ***distance function*** or ***metric*** if

- $D(p,q) \geq 0$ ($D(p,q) = 0$ if and only if $p = q$)
- $D(p,q) = D(q,p)$
- $D(p,z) \leq D(p,q) + D(q,z)$

Euclidean Distance

The Euclidean distance between the pixels p and q, with coordinates (x, y) and (s, t), respectively, can be defined as

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

The advantage of the Euclidean distance is its simplicity. However, since its calculation involves a square root operation, it is computationally very costly.

D4-distance (*city-block distance*)

D_4-distance (*city-block distance*) is defined as

$$D_4(p,q) = |x-s| + |y-t|$$

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

Pixels with $D_4(p) = 1$ is 4-neighbors of p.

D8-distance (*chessboard distance*)

D_8-distance (*chessboard distance*) is defined as

$$D_8(p,q) = \max(|x-s|, |y-t|)$$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Pixels with $D_8(p) = 1$ is 8-neighbors of p.

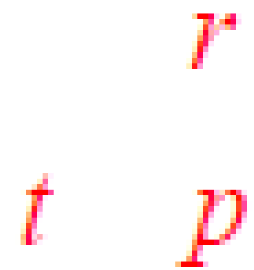
Exercise

- Compute D_e, D_4, D_8

0	1	1	1 (z)
1	0	0	1
1	1	1	1 (q)
1	1	1	1
(p)			

Labeling of Connected Components

- Scan the image from left to right
- Let p denote the pixel at any step in the scanning process.
- Let r denote the upper neighbor of p.
- Let t denote the left-hand neighbors of p, respectively.
- when we get to p, points r and t have already been encountered and labeled if they were 1's.



- if the value of p = 0, move on.
- if the value of p = 1, examine r and t.
  - if they are both 0, assign a new label to p.
  - if only one of them is 1, assign its label to p.
  - if they are both 1
    - if they have the same label, assign that label to p.
    - if not, assign one of the labels to p and make a note that the two labels are equivalent. (r and t are connected through p).

- at the end of the scan, all points with value 1 have been labeled.
- do a second scan, assign a new label for each equivalent labels.

What shall we do with 8-connected components ?

- do the same way but examine also the upper diagonal neighbors of p.
  - if p is 0, move on.
  - if p is 1
  - if all four neighbors are 0, assign a new label to p.
  - if only one of the neighbors is 1, assign its label to p.
  - if two or more neighbors are 1, assign one of the label to p and make a note of equivalent classes.
  - after complete the scan, do the second round and introduce a unique label to each equivalent class

Classification of Image Operations

- One way of classification is
  - Point
  - Local and
  - Global

- Point operation are those whose output value at a specific coordinate is dependent only on the input value.

- A local operation is one whose output value at a specific coordinate is dependent on the input values in the neighbourhood at that pixel.

- Global operation are those whose output value at a specific coordinate is dependent on all the values in the input image.

1. Linear operations 2. Non-linear operations

An operator is called a linear operator if it obeys the following rules of additivity and homogeneity. A non-linear operator, as the name suggests, does not follow these rules.

1. Property of additivity

$$H(a_1 f_1(x, y) + a_2 f_2(x, y)) = H(a_1 f_1(x, y)) + H(a_2 f_2(x, y))$$
$$= a_1 H(f_1(x, y)) + a_2 H(f_2(x, y))$$
$$= a_1 \times g_1(x, y) + a_2 \times g_2(x, y)$$

2. Property of homogeneity

$$H(k f_1(x, y)) = kH(f_1(x, y)) = kg_1(x, y)$$

Image Vs Array Operations

Image operations are array operations. These operations are done on a pixel-by-pixel basis. Array operations are different from matrix operations. For example, consider two images

$$F_1 = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \text{ and } F_2 = \begin{pmatrix} E & F \\ G & H \end{pmatrix}$$

The multiplication of F_1 and F_2 is element-wise, as follows:

$$F_1 \times F_2 = \begin{pmatrix} AE & BF \\ CG & HD \end{pmatrix}$$

In addition, one can observe that $F_1 \times F_2 = F_2 \times F_1$, whereas matrix multiplication is clearly different, since in matrices, $A \times B \neq B \times A$. By default, image operations are array operations only.

Arithmetic operations - Addition

Two images can be added in a direct manner, as given by

$$g(x, y) = f_1(x, y) + f_2(x, y)$$

Table 3.1 Data type and allowed ranges

S. no.	Data type	Data range
1	Uint 8	0–255
2	Uint 16	0–65,535
3	Uint 32	0–4,29,49,67,295
4	Uint 64	0–1,84,46,74,40,73,70,95,51,615

Similarly, it is possible to add a constant value to a single image, as follows:

$$g(x, y) = f_1(x, y) + k$$

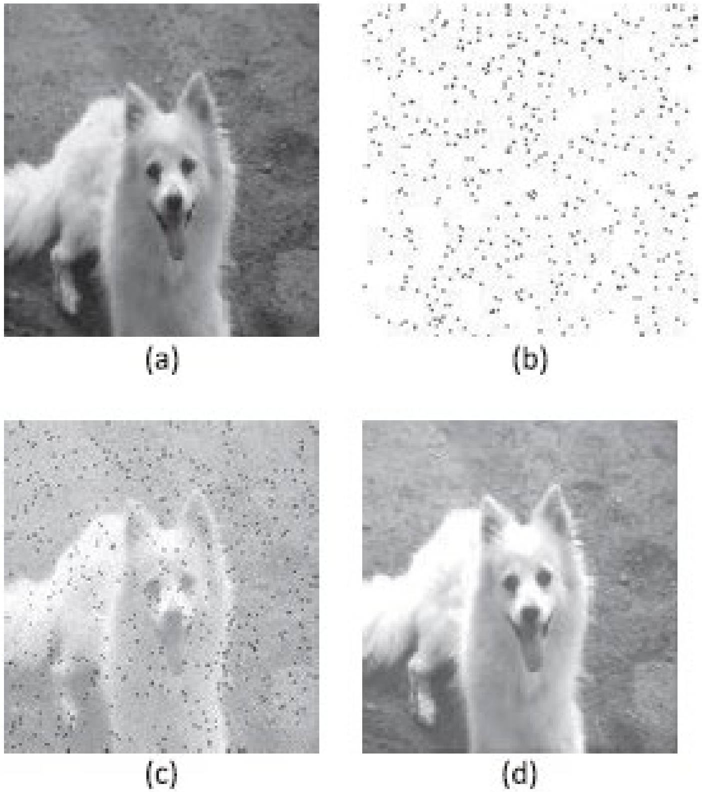


Fig. 3.14 Results of the image addition operation (a) Image 1 (b) Image 2 (c) Addition of images 1 and 2 (d) Addition of image 1 and constant 50

- Application of Image Addition
  - To increase the brightness of an image.
  - To create double exposure
    - Double exposure is he technique of superimposing an image on another image to produce the resultant. This gives a scenario equivalent to exposing a film to two picture.

Image Subtraction

The subtraction of two images can be done as follows. Consider

$$g(x, y) = f_1(x, y) - f_2(x, y)$$

where $f_1(x, y)$ and $f_2(x, y)$ are two input images and $g(x, y)$ is the output image. To avoid negative values, it is desirable to find the modulus of the difference as

$$g(x, y) = \left| f_1(x, y) - f_2(x, y) \right|$$

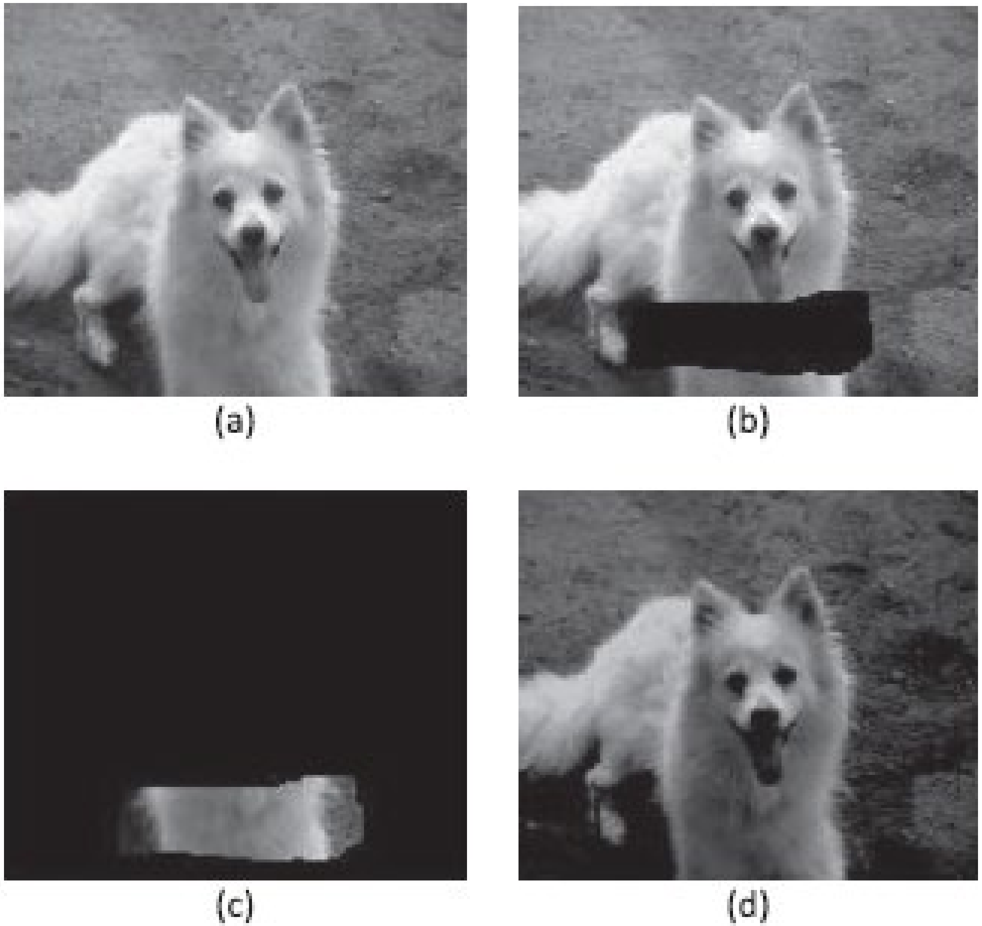


Fig. 3.15 Results of the image subtraction operation (a) Image 1 (b) Image 2 (c) Subtraction of images 1 and 2 (d) Subtraction of constant 50 from image 1

- Applications of Image Subtraction
  - Background elimination
  - Brightness reduction
  - Change detection
    - Single change detection
    - Double change detection

Image Multiplication

$$g(x, y) = f_1(x, y) \times f_2(x, y)$$

$$g(x, y) = f(x, y) \times k$$

Fig. 3.16 Result of multiplication operation (image × 1.25) resulting in good contrast

- Applications of Image Multiplication
  - It increases contrast
  - It is useful for designing filter masks.

Image Division

Similar to the other operations, division can be performed as

$$g(x, y) = \frac{f_1(x, y)}{f_2(x, y)}$$

where $f_1(x, y)$ and $f_2(x, y)$ are two input images and $g(x, y)$ is the output image.

$$g(x, y) = \frac{f(x, y)}{k}, \text{ where } k \text{ is a constant.}$$

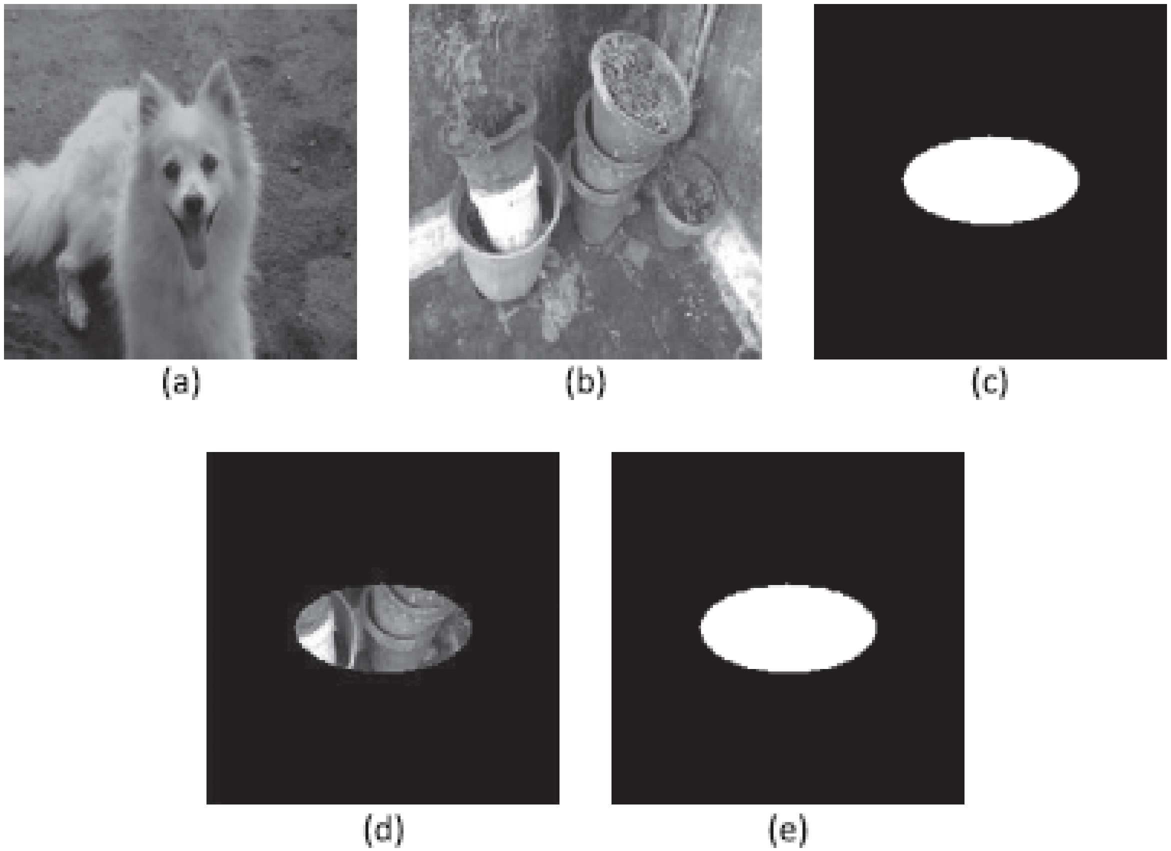


Fig. 3.17 Image division operation (a) Result of the image division operation (image/1.25) (b) Image 1 (c) Image 2 used as a mask (d) Image 3 = image 1 × image 2 (e) Image 4 = image 3/image 1

- Applications of Image division
  - Change detection
  - Contrast reduction

Logical Operations

1. AND/NAND
2. OR/NOR
3. EXOR/EXNOR
4. Invert/Logical NOT

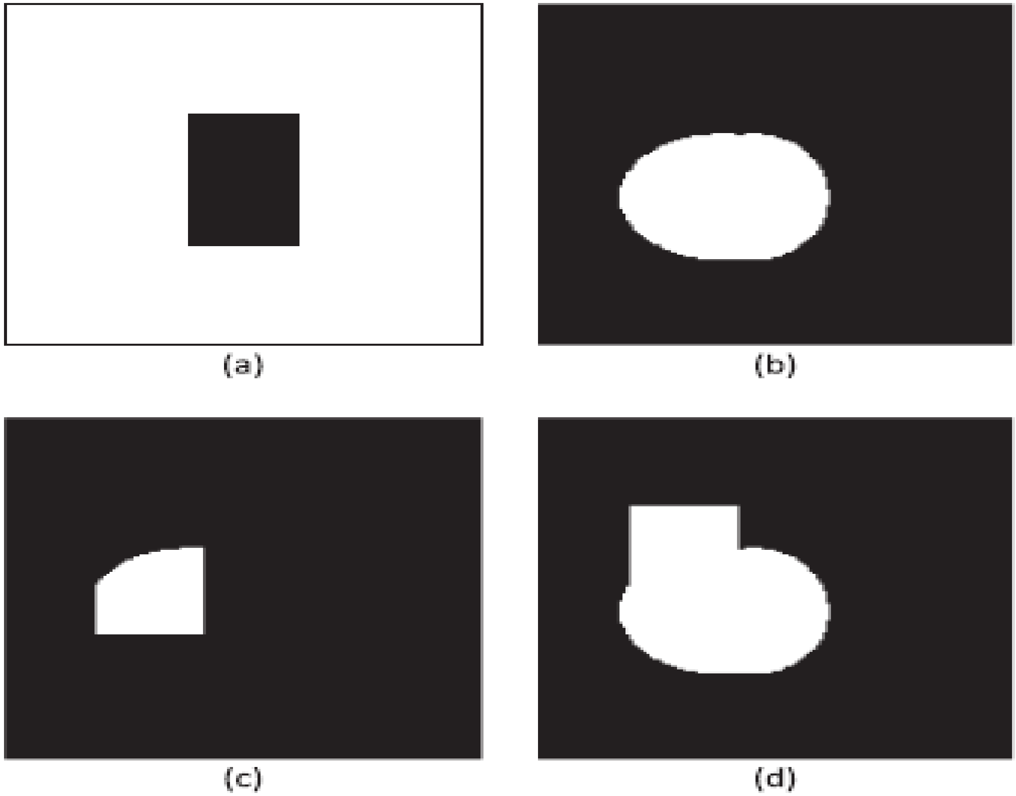


Fig. 3.18 Results of the AND and OR logical operators (a) Image 1 (b) Image 2 (c) Result of image 1 AND image 2 (d) Result of image 1 OR image 2

XOR

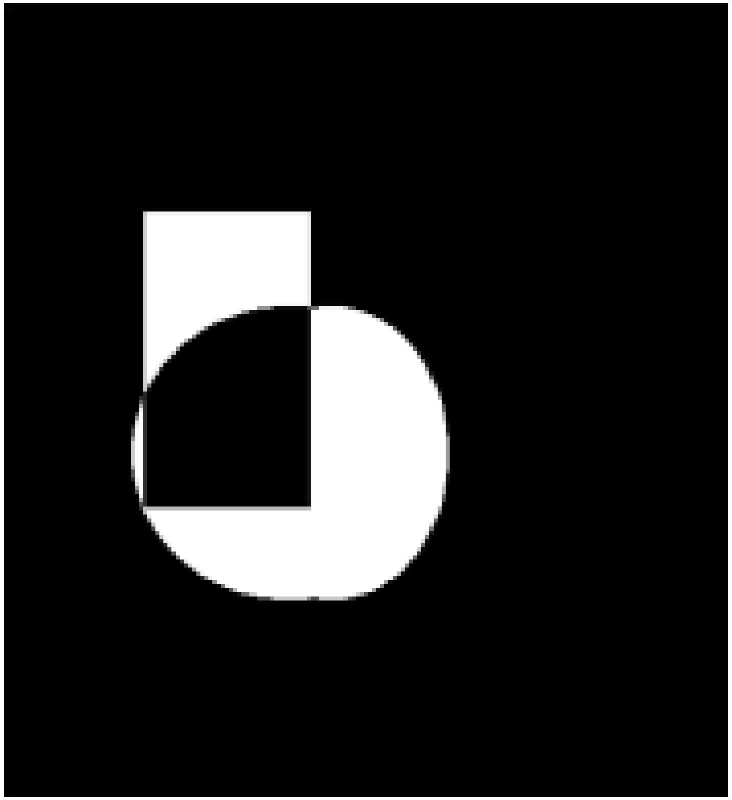


Fig. 3.19 Result of the XOR operation

NOT Operation

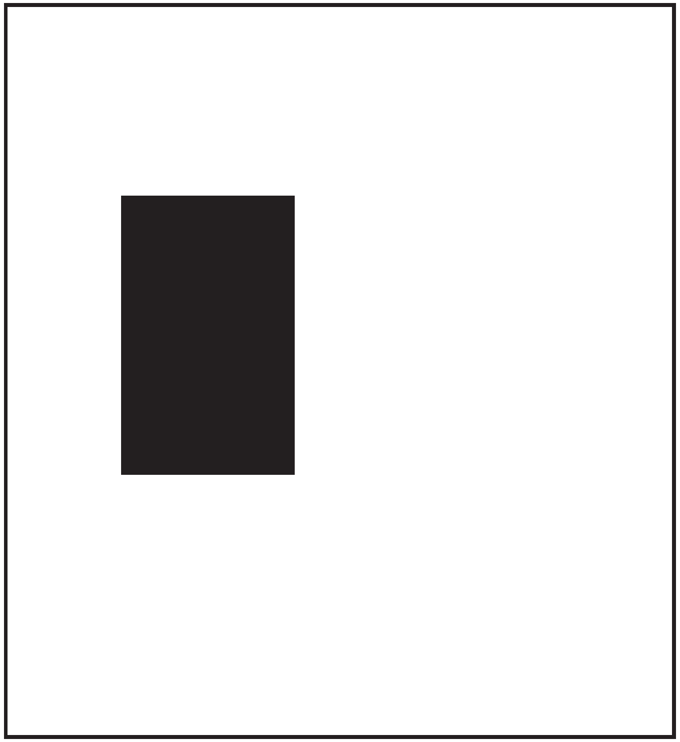
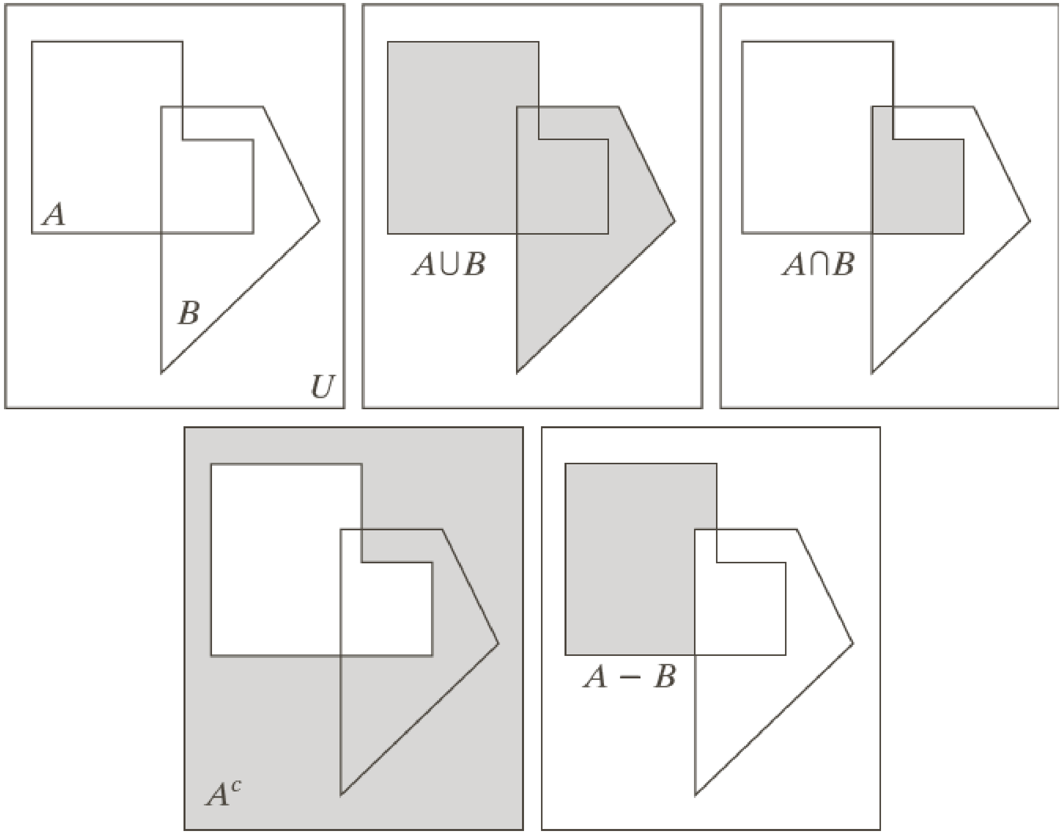


Fig. 3.20 Result of the NOT operation



a b c
d e

FIGURE 2.31
(a) Two sets of coordinates, A and B, in 2-D space. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B. In (b)–(e) the shaded areas represent the member of the set operation indicated.

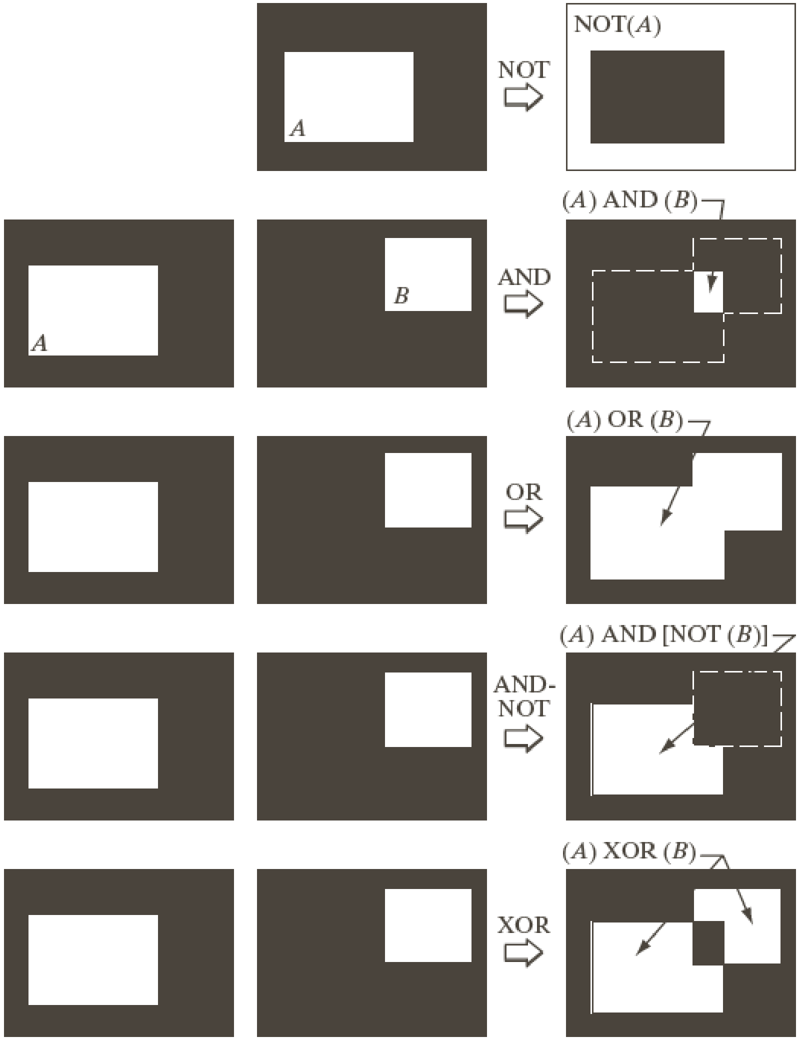
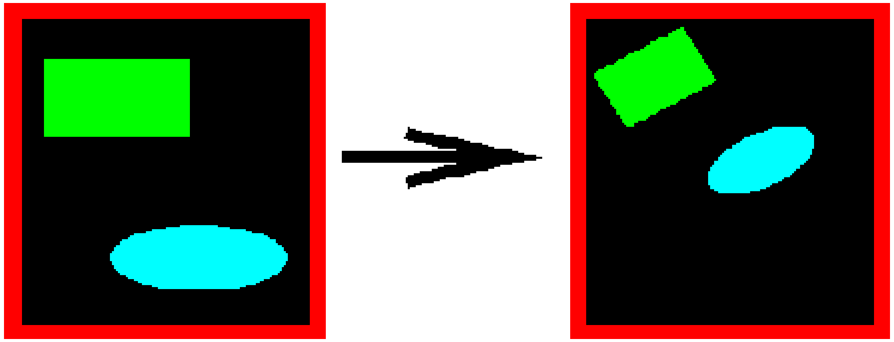


FIGURE 2.33
Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Geometrical Operation

- Translation
- Scaling
- Zooming
- Shearing
- Rotation
- Inverse Transform



Translation

- Translation is the movement of an image to a new position.

$$x' = x + \delta x$$

$$y' = y + \delta y$$

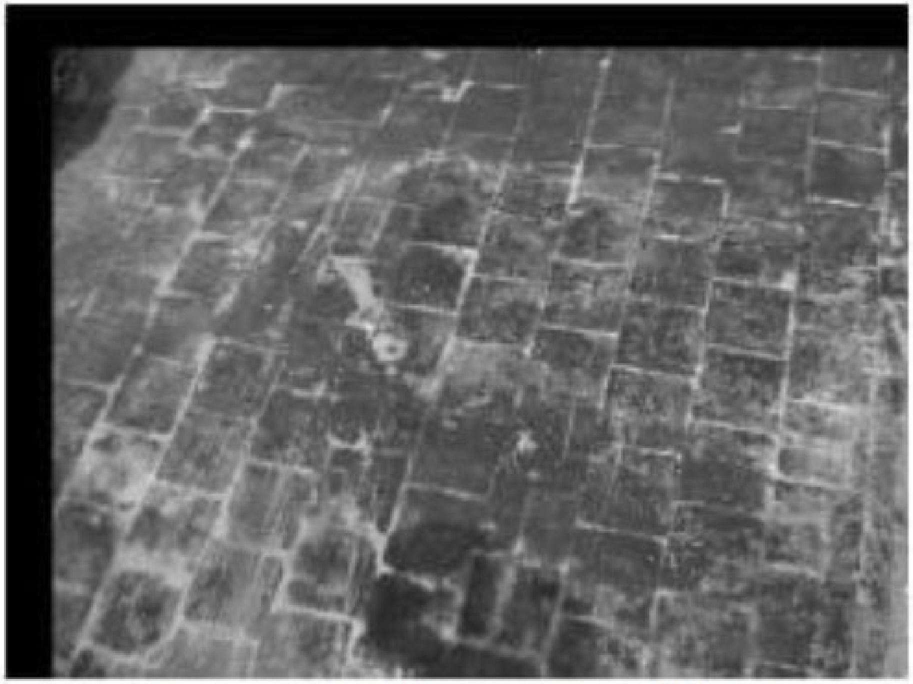


Fig. 3.21 Result of translation by 50 units

Scaling

- Scaling means enlarging and shrinking

$$x' = x \times Sx$$
$$y' = y \times Sy$$

$$[x', y'] = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} [x, y]$$

$$[x', y', 1] = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} [x, y, 1]^T$$

The matrix $S = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is called scaling matrix.

Zooming

For example, the image F is replicated as follows:

$$\begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} =$$

2	0	1	0
0	0	0	0
1	0	3	0
0	0	0	0

Linear Interpolation

Consider the image

$$H = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$$

Linear interpolation is equivalent to fitting a straight line by taking the average along the rows and the columns. The process is described as follows:

1. For example, the matrix H can be zero-interlaced as

$H =$

2	0	1	0
0	0	0	0
1	0	3	0
0	0	0	0

2. Interpolate the rows. This is achieved by taking the average of the columns. This yields

2	1.5	1	0.5
0	0	0	0
1	2	3	1.5
0	0	0	0

3. Interpolate the columns. This is achieved by taking the average of the rows. This yields

2	1.5	1	0.5
1.5	1.75	2	1
1	2	3	1.5
0.5	1	1.5	0.75

Reflection

Reflection along X

$$F' = [-x, y] = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \times [x, y]^T$$

Similarly, the reflection along the y-axis is given by

$$F' = [x, -y] = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \times [x, y]^T$$

Similarly, the reflection about the line $y = x$ is given as

$$F' = [x, -y] = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \times [x, y]^T$$

The reflection about $y = -x$ is given as

$$F' = [x, -y] = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \times [x, y]^T$$

Shearing

- Shearing is a transformation that produces a distortion of shape.

Shearing can be done using the following calculation and can be represented in the matrix form as

$$x' = sh_x \times y$$
$$y' = y$$
$$X_{shear} = \begin{pmatrix} 1 & 0 & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Similarly, Y_{shear} can be given as

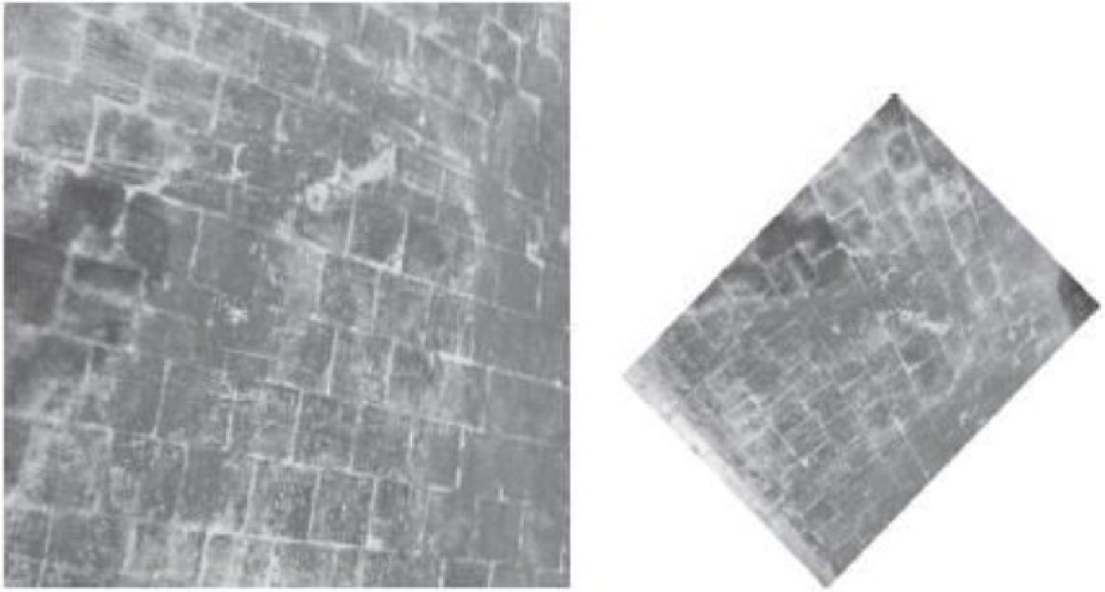
$$x' = x$$
$$y' = y \times sh_y$$
$$Y_{shear} = \begin{pmatrix} 1 & sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where sh_x and sh_y are shear factors in the x and y directions, respectively.

Rotation

$$[x', y', 1] = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} [x, y, 1]^{T}$$

If θ is substituted with $-\theta$, this matrix rotates the image in the clockwise direction.



(a) (b)

Fig. 3.23 Rotation (a) Original image (b) Result of rotation by 45°

Affine Transform

$$x' = T_x(x, y)$$

$$y' = T_y(x, y)$$

T_x and T_y are expressed as polynomials. The linear equation gives an affine transform.

$$x' = a_0 x + a_1 y + a_2$$

$$y' = b_0 x + b_1 y + b_2$$

This is expressed in matrix form as

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Inverse Transform

$$\text{Inverse transform for translation} = \begin{pmatrix} 1 & 0 & -\delta x \\ 0 & 1 & -\delta y \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{Inverse transform for scaling} = \begin{pmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Inverse transform for rotation can be obtained by changing the sign of the transform term. For example, the following matrix performs inverse transform.

$$\begin{pmatrix} \cos\theta & +\sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$