DIP NOTES

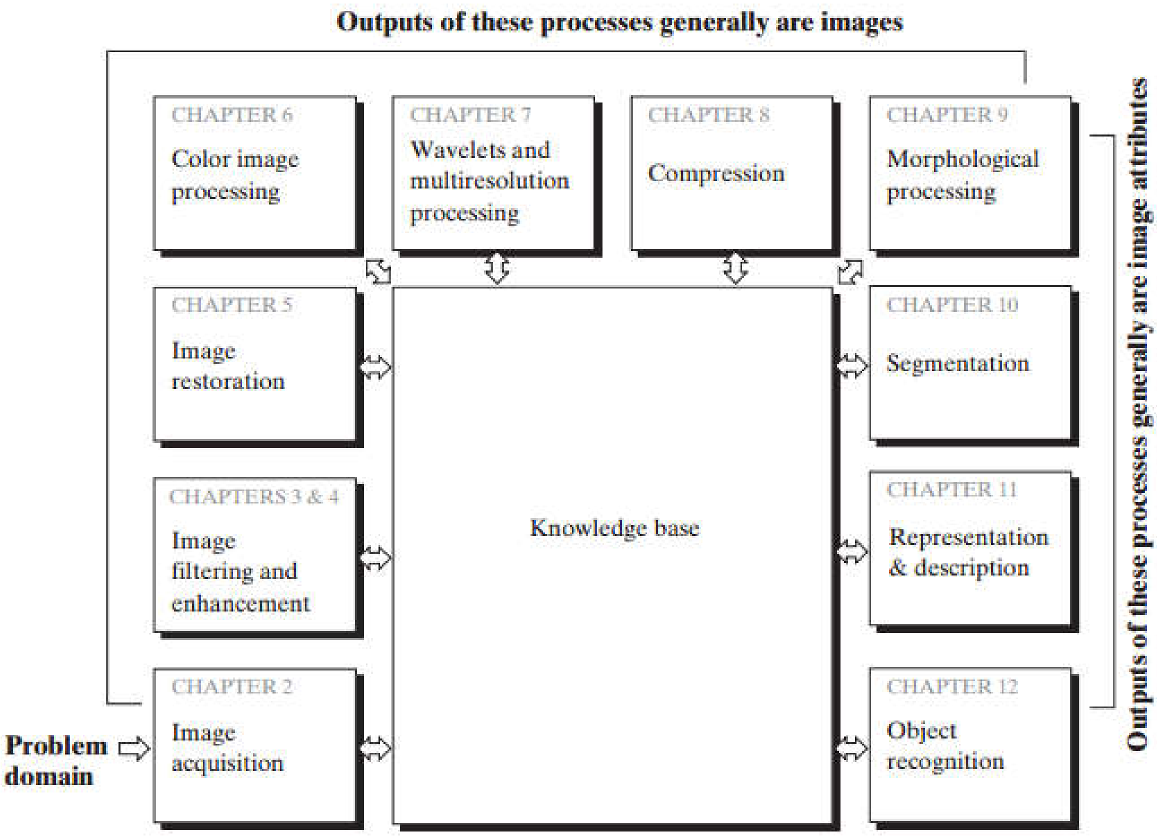
Based on previous year questions

Chapter – 1

1. Examples of Fields that Use Digital Image Processing

    a. Gamma Ray Imaging
    b. X Ray Imaging
    c. Imaging in Ultra violet band, Visible Band, Infrared Bands
    d. Imaging in Microwave Band, Radio Band

2. Fundamental Steps in Digital Image Processing – diagram



Chapter – 2

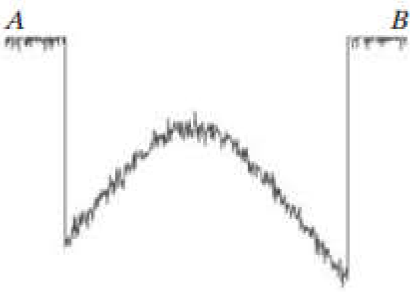
1. explain sampling and quantization

   The output of most sensors is a continuous voltage waveform, i.e. an analog image. To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes: sampling and quantization.

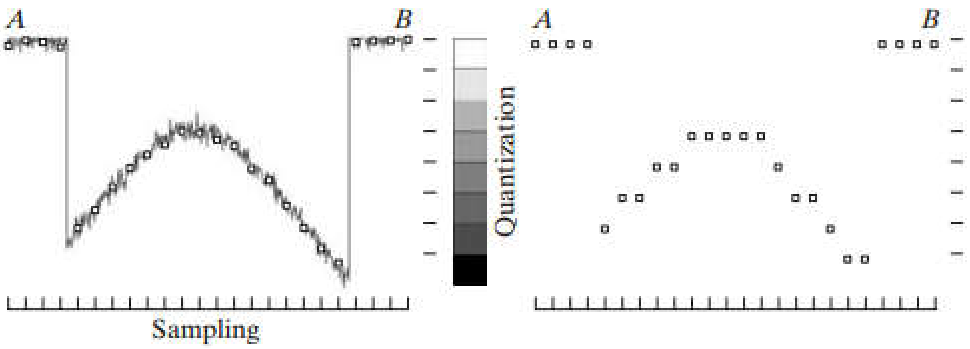
   An image may be continuous with respect to the x- and y-coordinates, and also in amplitude. To convert it to digital form, we have to sample the function in both coordinates and in amplitude/intensity.

   Digitizing the coordinate values is called **sampling**.
   Digitizing the amplitude/intensity values is called **quantization.**

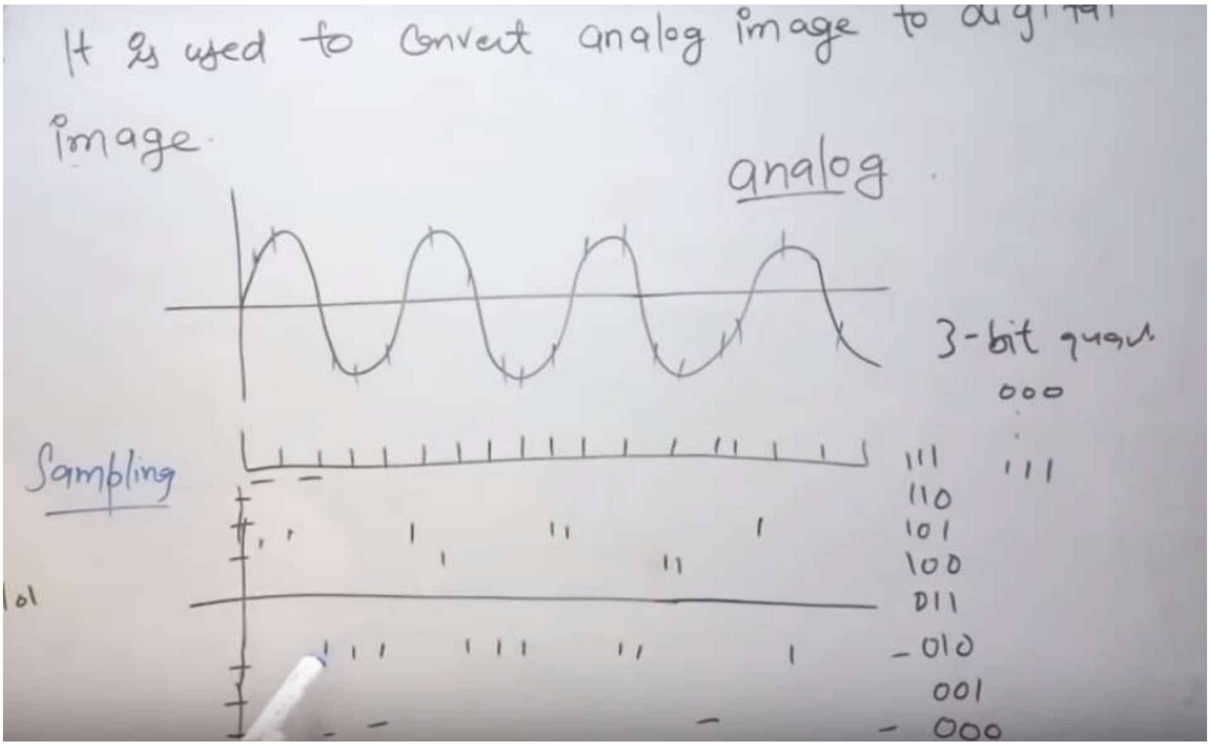
   Assume that this is the analog image,



   In sampling, we find coordinates, as done in the left side.
   In quantization, we took 3-bit quantization here, so there are 8 intensity values. We assign these from top to bottom.



See this, as well



2. What should be the properties of distance function/ metrics?

For pixels p, q, and z, with coordinates (x, y), (s, t), and (v, w), respectively, D is a *distance function* or *metric* if

(a) $D(p, q) \geq 0 \quad (D(p, q) = 0 \quad \text{iff} \quad p = q)$,
(b) $D(p, q) = D(q, p)$, and
(c) $D(p, z) \leq D(p, q) + D(q, z)$.

3. Consider image and find Euclidean, chess, and city block distance between pixel p and q

Concepts –

For pixels p, q, and z, with coordinates (x, y), (s, t), and (v, w), respectively,

The *Euclidean distance* between p and q is defined as

$$D_e(p, q) = \left[(x - s)^2 + (y - t)^2\right]^{\frac{1}{2}} \tag{2.5-1}$$

The D_4 *distance* (called the *city-block distance*) between p and q is defined as

$$D_4(p, q) = |x - s| + |y - t| \tag{2.5-2}$$

The D_8 *distance* (called the *chessboard distance*) between p and q is defined as

$$D_8(p, q) = \max(|x - s|, |y - t|) \tag{2.5-3}$$

Let the given image be –

4	2	2	2	(q)
3	3	1	3	
2	3	2	2	
2	1	2 p.	3	

We have to find the distance between p(3,2) and q(0,3)
P is (x, y), and q is (s ,t)

Euclidean distance, $D = \sqrt{(3-0)^2 + (2-3)^2}$

$= \sqrt{9+1} = \sqrt{10}$.

City-block distance $D_4(p,q)$

p is (3,2) and q is (0,3).

$D_4(p,q) = |3-0| + |2-3|$

$= 3+1 = 4$.

chessboard distance is ::

$D_8(p,q) = \max(|3-0|, |2-3|)$

$= \max(3,1)$

$= 3$.

4. Exercise question – 2.11

★2.11 Consider the two image subsets, S_1 and S_2, shown in the following figure. For $V = \{1\}$, determine whether these two subsets are (a) 4-adjacent, (b) 8-adjacent, or (c) m-adjacent.

		S_1				S_2			
0	0	0	0	0	0	0	1	1	0
1	0	0	1	0	0	1	0	0	1
1	0	0	1	0	1	1	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	0	0	1	1	1

First, we understand the meaning of 4-adjacent, 8-adjacent and m-adjacent

4-adjacent

A pixel p at coordinates (x, y) has four *horizontal* and *vertical* neighbors whose coordinates are given by

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$$

This set of pixels, called the 4-*neighbors* of p, is denoted by $N_4(p)$. Each pixel is a unit distance from (x, y), and some of the neighbor locations of p lie outside the digital image if (x, y) is on the border of the image. We deal with this issue

8-adjacent

The four *diagonal* neighbors of p have coordinates

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

and are denoted by $N_D(p)$. These points, together with the 4-neighbors, are called the 8-*neighbors* of p, denoted by $N_8(p)$. As before, some of the neighbor locations in $N_D(p)$ and $N_8(p)$ fall outside the image if (x, y) is on the border of the image.

(a) 4-*adjacency*. Two pixels p and q with values from V are 4-adjacent if q is in the set $N_4(p)$.

(b) 8-*adjacency*. Two pixels p and q with values from V are 8-adjacent if q is in the set $N_8(p)$.

(c) m-*adjacency* (mixed adjacency). Two pixels p and q with values from V are m-adjacent if

(i) q is in $N_4(p)$, *or*

(ii) q is in $N_D(p)$ *and* the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V.

Now coming to the original problem, determine whether two subsets are 4, 8, or m-adjacent. For this, we take 2 pixels anywhere in the sets.

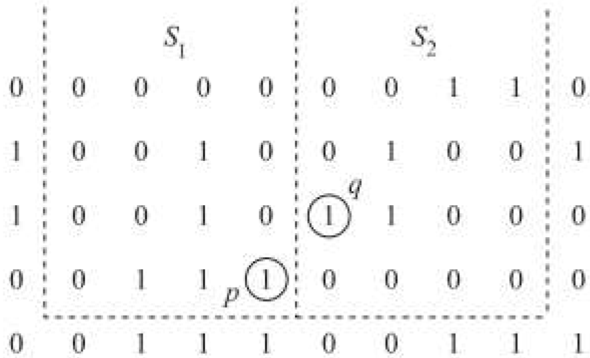


Figure P2.11

Let p and q be as shown in Fig. P2.11. Then, (a) S_1 and S_2 are not 4-connected because q is not in the set $N_4(p)$; (b) S_1 and S_2 are 8-connected because q is in the set $N_8(p)$; (c) S_1 and S_2 are m-connected because (i) q is in $N_D(p)$, and (ii) the set $N_4(p) \cap N_4(q)$ is empty.

Chapter – 3

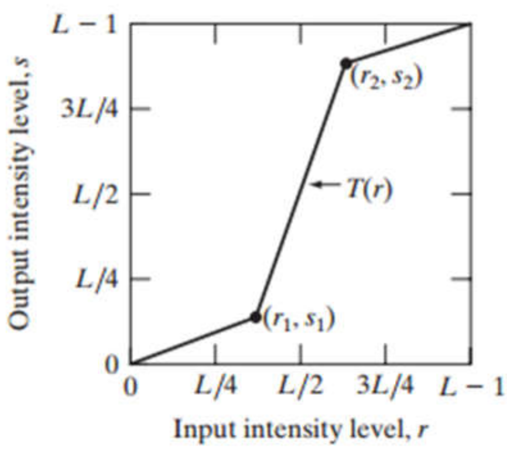
1. Piecewise linear transformation functions?

   Piecewise linear functions are majorly intensity transformation functions.
   There are three major types of linear transformation functions -

   a. Contrast stretching
   b. Intensity-level slicing
   c. Bit-plane slicing

   Contrast Stretching –
   Contrast stretching is a process that expands the range of intensity levels in an image so that it spans the full intensity range of the recording medium or display device.

   The following graph shows a typical transformation used for contrast stretching.

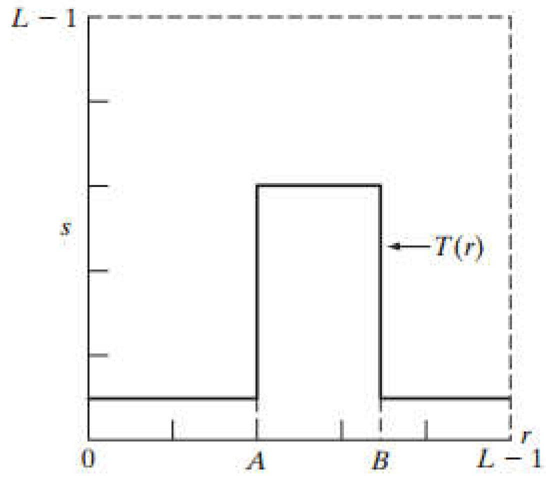


   The locations of points (r1, s1) and (r2, s2) control the shape of the transformation function.
   If r1 = s1 and r2 = s2, the transformation is a linear function that produces no changes in intensity levels.
   If r1 = r2, s1 = 0 and s2 = L – 1, the transformation becomes a thresholding function that creates a binary image.

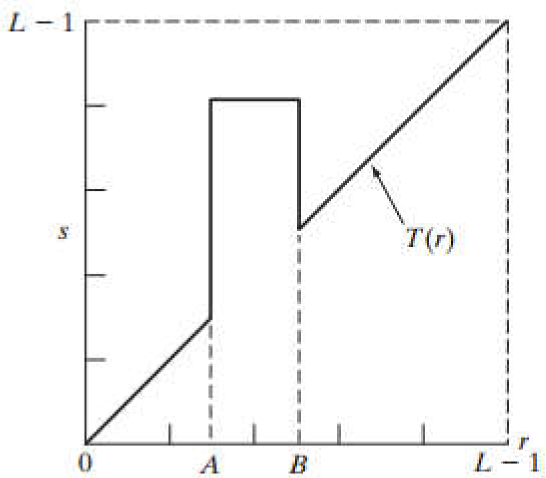
   In general, r1 <= r2 and s1 <= s2 is assumed so that the function is single valued and monotonically increasing. This condition preserves the order of intensity levels, thus preventing the creation of intensity artifacts in the processed image.

   Intensity-Level Slicing
   Highlighting a specific range of intensities in an image is called intensity-level.
   There are majorly two variations of this –

   One approach is to display in one value (say, white) all the values in the range of interest and in another (say, black) all other intensities. This transformation produces a binary image.
   In this, the background is destroyed.



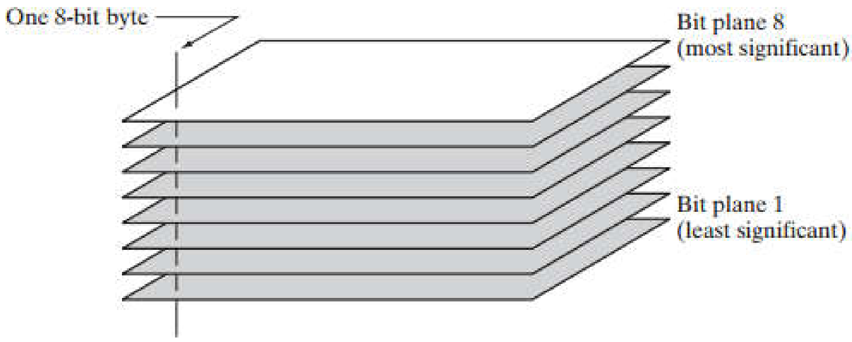
The second approach, based on the following transformation, brightens (or darkens) the desired range of intensities but leaves all other intensity levels in the image unchanged. In this, the background is preserved.



Bit-plane Slicing

Pixels are digital numbers composed of bits. For example, the intensity of each pixel in a 256-level gray-scale image is composed of 8 bits. Instead of highlighting intensity-level ranges, we could highlight the contribution made to total image appearance by specific bits.

An 8-bit image may be considered as being composed of eight 1-bit planes, with plane 1 containing the lowest-order bit of all pixels in the image and plane 8 all the highest-order bits.



In terms of intensity transformation functions, it is not difficult to show that the binary image for the 8th bit plane of an 8-bit image can be obtained by processing the input image with a thresholding intensity transformation function that maps all intensities of range 0-127 to 0 and all intensities of range 128-255 to 1.

Decomposing an image into its bit planes is useful for analysing the relative importance of each bit in the image. Also, this type of decomposition is useful for image compression, in which fewer than all planes are used in reconstructing an image.

2. What is a histogram? Explain histogram equalization. Given an input image, find the intensity values for equalized histogram.

Histogram -
The histogram of a digital image with intensity levels in the range [0, L - 1] is a discrete function h(rk) = nk, where rk is the kth intensity value and nk is the number of pixels in the image with intensity rk.
That is,
The histogram of an image is a plot of the number of occurrences of gray levels in the image against the gray level values.

Histogram Equalization –
Equalization is a process that attempts to spread out the gray levels in an image so that they are evenly distributed across their image.
Histogram equalization is a technique where the histogram of the resultant image is as flat as possible.
Related video - https://youtu.be/fGymRwRjylo

Steps to perform histogram equalization are –
a. Make a table of gray levels, and frequency
b. Find PDF (probability density function) i.e. divide each value of intensity by sum of intensities. The sum of this column must be 1.
c. Find CDF (cumulative density function) i.e. do cumulative sum of PDF.
d. Multiply the results by maximum gray level value (7 for 3-bit image levels)
e. Round the result to closest integer (>0.5 goes to ceil and < 0.5 is floor)

This gives us one-to-one correspondence in input and this result, i.e. we can replace the input gray levels in the images by the corresponding result value.

We can find intensity table for this equalized image for looking at the results. See the following example calculation.

Gray level	No of pixels at this gray level	PDF No of pixels/ total	CDF	Multiply by 3	Result (dummy)
0	10				0
1	12				0
2	20				2
3	13				3

Now we look at result and make the following table.
The equalized histogram table will be as follows –

Gray levels	No of pixels at this gray level
0	10 + 12 = 22
1	20
2	0
3	13

Now plot the histogram for this new image using these values.
This is the equalized histogram.

Histogram Equalization example – An actual numerical would be solved like this

Gray level	No of pixels at this gray level	PDF No of pixels/ total	CDF	Multiply by 3 i.e. max gray level	Result
0	10	10/70 = 0.14285	0.14285	0.4285	0
1	15	15/70 = 0.21428	0.35713	1.0714	1
2	20	20/70 = 0.28571	0.64284	1.9285	2
3	25	25/70 = 0.35714	0.99998	2.9999	3

3. **What is Histogram Matching?** Given two image intensity distributions. It is desirable to transform the image A to B. Show the intensity distribution of transformed image using histogram matching.

Image A

Gray level	0	1	2	3	4	5	6	7
No of pixels	8	10	10	2	12	16	4	2

Image B

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	0	20	20	16	8

Histogram Matching (Specification) –

It is sometimes desirable to be able to specify the shape of the histogram that we wish the processed image to have. The method used to generate a processed image that has a specified histogram is called histogram matching or histogram specification.
Related video - https://youtu.be/nwIT4HJrKXs

Steps for histogram matching –

a. Find the CDF of both distributions, T(r) for A, and G(z) for B
b. Match gray levels such that -> cdf(B) >= cdf(A)
c. Now we have one-to-one correspondence
d. Generate new intensity table and draw histogram

In the above example, Let T(r) be CDF of A, and G(z) be CDF of B

Image A

Gray level	0	1	2	3	4	5	6	7
No of pixels	8	10	10	2	12	16	4	2
T(r) i.e. CDF for A	8	18	28	30	42	58	62	64

Image B

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	0	20	20	16	8
G(z) i.e. CDF for B	0	0	0	0	20	40	56	64

Now CDFs are -

Gray level	0	1	2	3	4	5	6	7
T(r) i.e. CDF for A	8	18	28	30	42	58	62	64
G(z) i.e. CDF for B	0	0	0	0	20	40	56	64

Now we perform matching such that, cdf(B) >= cdf(A) . That means, find a CDF in B, that is greater or equal to CDF in A, Thus the matching is –

Matching -

Gray level of A	0	1	2	3	4	5	6	7
Gray level of B	4	4	5	5	6	6	7	7

New Intensity table is –

Transformed Image A

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	0	8 + 10	10 + 2	12 + 16	4 + 2

Now simply draw histogram for this transformed image. Similar example in minor-1.

4. What is an order statistic filter? What will be the intensity of center pixel after applying the median filter in 3x3 neighbourhood?

10	20	20
20	25	20
15	15	30

Order statistic filter –

Order-statistic filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

The best-known filter in this category is the median filter, which, as its name implies, **replaces the value of a pixel by the median of the intensity values in the neighbourhood of that pixel** (the original value of the pixel is included in the computation of the median)

Median filters are popular because –

a. For certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size.
b. Median filters are particularly effective in the presence of impulse noise. (also called salt-and-pepper noise because of its appearance as white and black dots superimposed on an image).

In the given 3x3 matrix, to find the answer –

a. Sort the neighbours
b. Then simply find median

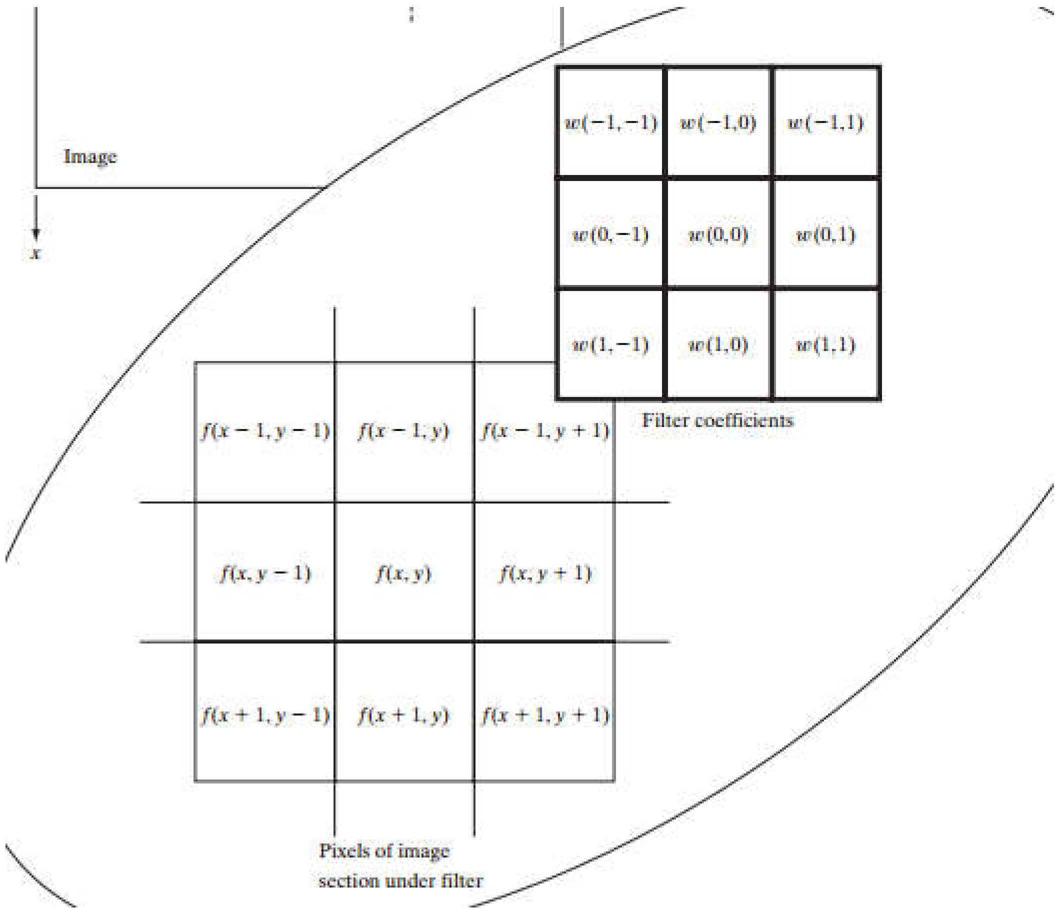
In our case, sorted values are – <u>10, 15, 15, 20</u>, **20**, <u>20, 20, 25, 30</u>
So the median is 20, So Intensity of center pixel will be 20.

Other types of order statistics filters are –
Min filters, Max filters, which do as the name suggests.

5. Mask processing techniques for image enhancement

Images can be enhanced by using masks or filters which are applied to the neighbourhood of the pixel. The value of each pixel in a grid is transformed using its neighbourhood and its corresponding masks. They are used for sharpening or blurring of images.

Let’s say the neighbourhood is 3x3. Mask is also 3x3 which contains weights of pixel value.



g(x,y) is known as the value of pixel (x,y) in transformed image.

$$g(x, y) = w(-1, -1)f(x-1, y-1) + w(-1, 0)f(x-1, y) + \ldots + w(0, 0)f(x, y) + \ldots + w(1, 1)f(x+1, y+1)$$

~~3 × 3.~~ In general, linear spatial filtering of an image of size $M \times N$ with a filter of size $m \times n$ is given by the expression:

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)f(x+s, y+t)$$

where x and y are varied so that each pixel in w visits every pixel in f.

Thus, by using masks, images can be enhanced.

1. For blurring, averaging filters (low pass) are used.
   a. Used in Noise reduction
   b. Used for smoothening of contours

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

a b

FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

2. Order statistics filters –

Order-statistic filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

The best-known filter in this category is the median filter, which, as its name implies, **replaces the value of a pixel by the median of the intensity values in the neighbourhood of that pixel** (the original value of the pixel is included in the computation of the median)

3. For sharpning, We use Laplacian mask which is derived by -

Therefore, it follows from the preceding three equations that the discrete Laplacian of two variables is

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \quad (3.6\text{-}6)$$

Example masks are –

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

FIGURE 3.37
(a) Filter mask used to implement Eq. (3.6-6).
(b) Mask used to implement an extension of this equation that includes the diagonal terms.
(c) and (d) Two other implementations of the Laplacian found frequently in practice.

sharpened result. Thus, the basic way in which we use the Laplacian for image sharpening is

$$g(x, y) = f(x, y) + c\left[\nabla^2 f(x, y)\right] \tag{3.6-7}$$

where $f(x, y)$ and $g(x, y)$ are the input and sharpened images, respectively. The constant is $c = -1$ if the Laplacian filters in Fig. 3.37(a) or (b) are used, and $c = 1$ if either of the other two filters is used.

6. Unsharp masking and high boost filtering in detail.

A process that has been used for many years by the printing and publishing industry to sharpen images consists of :
subtracting an unsharp (smoothed) version of an image from the original image.
This process, called unsharp masking, consists of the following steps:

1. Blur the original image.
2. Subtract the blurred image from the original (the resulting difference is called the mask.)
3. Add the mask to the original.

Letting $\bar{f}(x, y)$ denote the blurred image, unsharp masking is expressed in equation form as follows. First we obtain the mask:

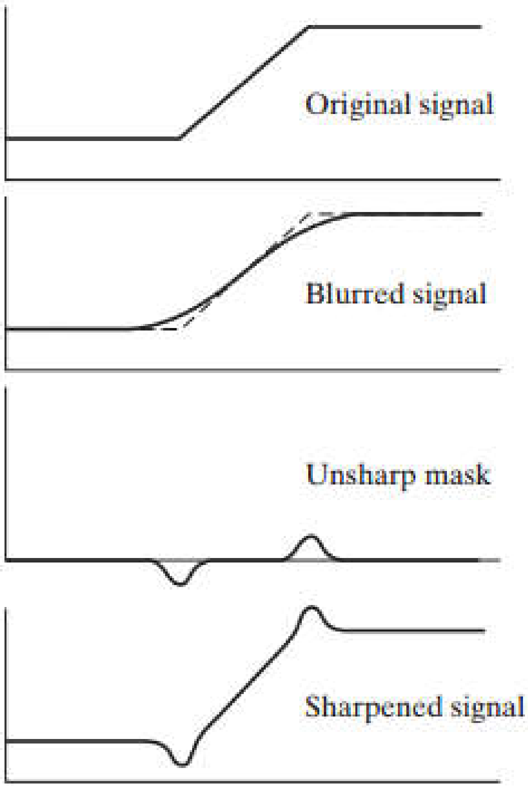
$$g_{\text{mask}}(x, y) = f(x, y) - \bar{f}(x, y) \tag{3.6-8}$$

Then we add a weighted portion of the mask back to the original image:

$$g(x, y) = f(x, y) + k * g_{\text{mask}}(x, y) \tag{3.6-9}$$

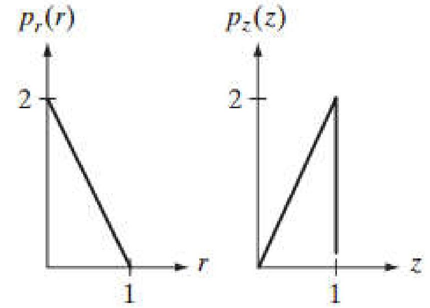
where we included a weight, k ($k \geq 0$), for generality. When $k = 1$, we have unsharp masking, as defined above. When $k > 1$, the process is referred to as *highboost filtering*. Choosing $k < 1$ de-emphasizes the contribution of the unsharp mask.

Diagram is as follows -


Original signal
Blurred signal
Unsharp mask
Sharpened signal

7. Exercise Question 3.11

3.11 An image with intensities in the range [0, 1] has the PDF $p_r(r)$ shown in the following diagram. It is desired to transform the intensity levels of this image so that they will have the specified $p_z(z)$ shown. Assume continuous quantities and find the transformation (in terms of r and z) that will accomplish this.



Weird question. Just learn it as it is.

Problem 3.11

First, we obtain the histogram equalization transformation:

$$s = T(r) = \int_0^r p_r(w)\,dw = \int_0^r (-2w+2)\,dw = -r^2 + 2r.$$

Next we find

$$v = G(z) = \int_0^z p_z(w)\,dw = \int_0^z 2w\ dw = z^2.$$

Finally,

$$z = G^{-1}(v) = \pm\sqrt{v}.$$

But only positive intensity levels are allowed, so $z = \sqrt{v}$. Then, we replace v with s, which in turn is $-r^2 + 2r$, and we have

$$z = \sqrt{-r^2 + 2r}.$$

Chapter – 4

1. Explain any four properties of 2D discrete Fourier transformation (DFT)

   There are many properties, some of them are –

   a. Symmetry

Spatial Domain†		**Frequency Domain**†
$f(x, y)$ real	$\Leftrightarrow$	$F^*(u, v) = F(-u, -v)$
$f(x, y)$ imaginary	$\Leftrightarrow$	$F^*(-u, -v) = -F(u, v)$

   b. Linearity

2) Linearity $\quad af_1(x, y) + bf_2(x, y) \Leftrightarrow aF_1(u, v) + bF_2(u, v)$

   c. Rotation

5) Rotation $\quad f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$

$x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$

   d. Convolution

6) Convolution theorem† $\quad f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$

$f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$

   e. Correlation theorem

7) Correlation theorem† $\quad f(x, y) \,☆\, h(x, y) \Leftrightarrow F^*(u, v)\,H(u, v)$

$f^*(x, y)h(x, y) \Leftrightarrow F(u, v) \,☆\, H(u, v)$

   f. Discrete unit impulse

8) Discrete unit impulse $\quad \delta(x, y) \Leftrightarrow 1$

2. What are the basic steps involved in filtering in frequency domain?

Related video - https://www.youtube.com/watch?v=fyoC6aqAgK4
Steps involved for filtering in frequency domain –

1. Given an input image $f(x, y)$ of size $M \times N$, obtain the padding parameters P and Q from Eqs. (4.6-31) and (4.6-32). Typically, we select $P = 2M$ and $Q = 2N$.
2. Form a padded image, $f_p(x, y)$, of size $P \times Q$ by appending the necessary number of zeros to $f(x, y)$.
3. Multiply $f_p(x, y)$ by $(-1)^{x+y}$ to center its transform.
4. Compute the DFT, $F(u, v)$, of the image from step 3.
5. Generate a real, symmetric filter function, $H(u, v)$, of size $P \times Q$ with center at coordinates $(P/2, Q/2)$.† Form the product $G(u, v) = H(u, v)F(u, v)$ using array multiplication; that is, $G(i, k) = H(i, k)F(i, k)$.
6. Obtain the processed image:

$$g_p(x, y) = \left\{\text{real}\left[\Im^{-1}[G(u, v)]\right]\right\}(-1)^{x+y}$$

   where the real part is selected in order to ignore parasitic complex components resulting from computational inaccuracies, and the subscript p indicates that we are dealing with padded arrays.
7. Obtain the final processed result, $g(x, y)$, by extracting the $M \times N$ region from the top, left quadrant of $g_p(x, y)$.

Simplified as,

Steps for filtering in frequency domain

1. $f(x,y)$. dimension $M \times N$
2. $f_p(x,y)$, dimension $P \times Q$, $P = 2M$ and $Q = 2N$.
3. $f_p(x,y) \times (-1)^{x+y}$ Mul Spatial Convolution
4. $H(u,v)$ - filter, $F(x,y)$

   $G(u,v) = H(u,v) \times F(x,y)$
5. $g(x,y) = \Im^{-1}[G(u,v)](-1)^{x+y}$

3. Filters - Low pass. High pass – Ideal, Butterworth, Gaussian

   Use this video - https://www.youtube.com/watch?v=2cQkm3e8KPk
   This covers all the filters without enough content.
   Draw diagrams as well.

Chapter – 9

1. Study erosion and dilation concepts and numericals.

Erosion

With A and B as sets in Z^2, the erosion of A by B, denoted $A \ominus B$, is defined as

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \tag{9.2-1}$$

$$A \ominus B = \{z \mid (B)_z \cap A^c = \varnothing\}$$

Erosion example –

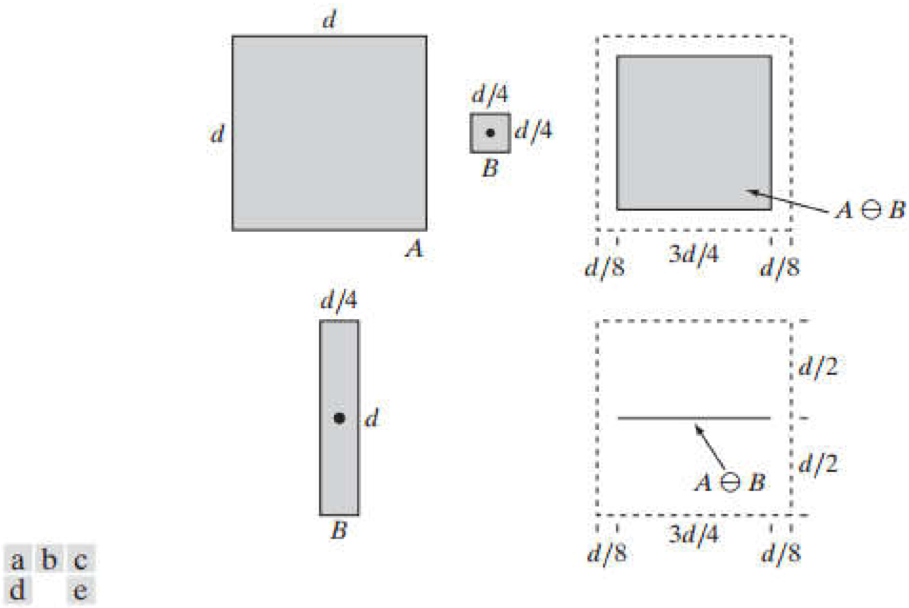


FIGURE 9.4 (a) Set A. (b) Square structuring element, B. (c) Erosion of A by B, shown shaded. (d) Elongated structuring element. (e) Erosion of A by B using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference.

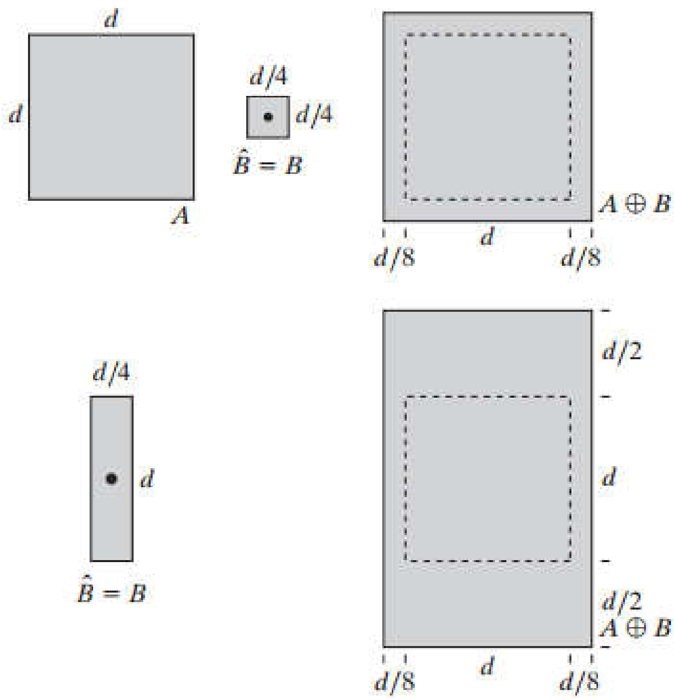
Dilation –

With A and B as sets in Z^2, the *dilation* of A by B, denoted $A \oplus B$, is defined as

$$A \oplus B = \left\{z \mid (\hat{B})_z \cap A \neq \varnothing\right\} \tag{9.2-3}$$

$$A \oplus B = \left\{z \mid [(\hat{B})_z \cap A] \subseteq A\right\}$$

Dilation example -



a b c
d e

FIGURE 9.6 (a) Set A. (b) Square structuring element (the dot denotes the origin). (c) Dilation of A by B, shown shaded. (d) Elongated structuring element. (e) Dilation of A using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference

Also,

9.2.3 Duality

Erosion and dilation are duals of each other with respect to set complementation and reflection. That is,

$$(A \ominus B)^c = A^c \oplus \hat{B} \tag{9.2-5}$$

and

$$(A \oplus B)^c = A^c \ominus \hat{B} \tag{9.2-6}$$

2. Opening and Closing morphological operations with their applications?

Opening - Opening generally smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions.

The *opening* of set A by structuring element B, denoted $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B \tag{9.3-1}$$

Thus, the opening A by B is the erosion of A by B, followed by a dilation of the result by B.

Closing – Closing also tends to smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour.

Similarly, the *closing* of set A by structuring element B, denoted $A \bullet B$, is defined as

$$A \bullet B = (A \oplus B) \ominus B \tag{9.3-2}$$

which says that the closing of A by B is simply the dilation of A by B, followed by the erosion of the result by B.

3. short note on Hit or Miss transformation

Related video - https://www.youtube.com/watch?v=4IKNhRjPIB8
The hit-and-miss transform is a general binary morphological operation that can be used to look for particular patterns of foreground and background pixels in an image. The morphological hit-or-miss transform is a basic tool for shape detection.

It is actually the basic operation of binary morphology since almost all the other binary morphological operators can be derived from it. As with other binary morphological operators it takes as input a binary image and a structuring element, and produces another binary image as output.

The equation for morphological hit and miss operation is -

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2) \tag{9.4-2}$$

Thus, set $A \circledast B$ contains all the (origin) points at which, simultaneously, B_1 found a match ("hit") in A and B_2 found a match in A^c. By using the definition

We are using a structuring element B1 associated with objects, and an element B2, associated with the background, based on an assumed definition that two or more objects are distinct only if they form disjoint (disconnected) sets.

4. Explain skeletonization morphological operation with example.

Skeletonization –
Skeletonization is a process for reducing foreground regions in a binary image to a skeletal remnant that largely preserves the extent and connectivity of the original region while throwing away most of the original foreground pixels.

The skeleton is useful because it provides a simple and compact representation of a shape that preserves many of the topological and size characteristics of the original shape.

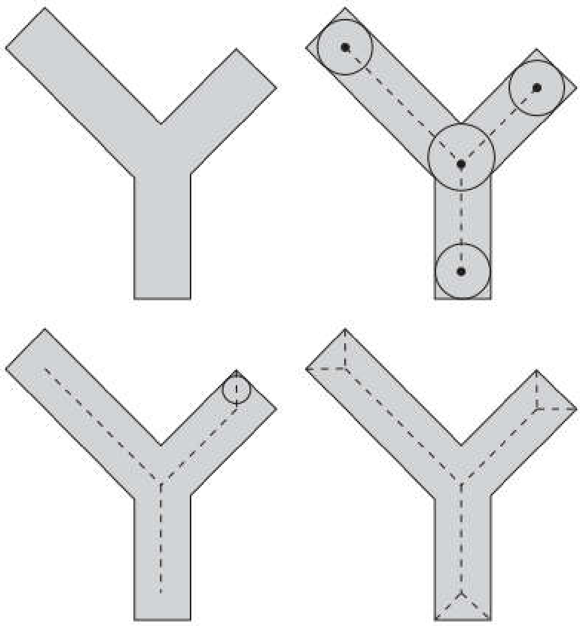
How to calculate skeleton –

Use some kind of morphological thinning that successively erodes away pixels from the boundary (while preserving the end points of line segments) until no more thinning is possible, at which point what is left approximates the skeleton.

As Fig. 9.23 shows, the notion of a skeleton, $S(A)$, of a set A is intuitively simple. We deduce from this figure that

(a) If z is a point of $S(A)$ and $(D)_z$ is the largest disk centered at z and contained in A, one cannot find a larger disk (not necessarily centered at z) containing $(D)_z$ and included in A. The disk $(D)_z$ is called a *maximum disk*.

(b) The disk $(D)_z$ touches the boundary of A at two or more different places.



a b
c d

FIGURE 9.23 (a) Set A. (b) Various positions of maximum disks with centers on the skeleton of A. (c) Another maximum disk on a different segment of the skeleton of A. (d) Complete skeleton.

In book method,

a. We use successive erosion, i.e. perform k-1 erosion where kth erosion will lead to empty set.
b. Image can be reconstructed from skeleton by applying successive dilations.

In terms of equation, we can write it as follows -

The skeleton of A can be expressed in terms of erosions and openings. That is, it can be shown (Serra [1982]) that

$$S(A) = \bigcup_{k=0}^{K} S_k(A) \tag{9.5-11}$$

with

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B \tag{9.5-12}$$

where B is a structuring element, and $(A \ominus kB)$ indicates k successive erosions of A:

$$(A \ominus kB) = ((\ldots((A \ominus B) \ominus B) \ominus \ldots) \ominus B) \tag{9.5-13}$$

k times, and K is the last iterative step before A erodes to an empty set. In other words,

$$K = \max\{k \mid (A \ominus kB) \neq \varnothing\} \tag{9.5-14}$$

The formulation given in Eqs. (9.5-11) and (9.5-12) states that $S(A)$ can be obtained as the union of the *skeleton subsets* $S_k(A)$. Also, it can be shown that A can be *reconstructed* from these subsets by using the equation

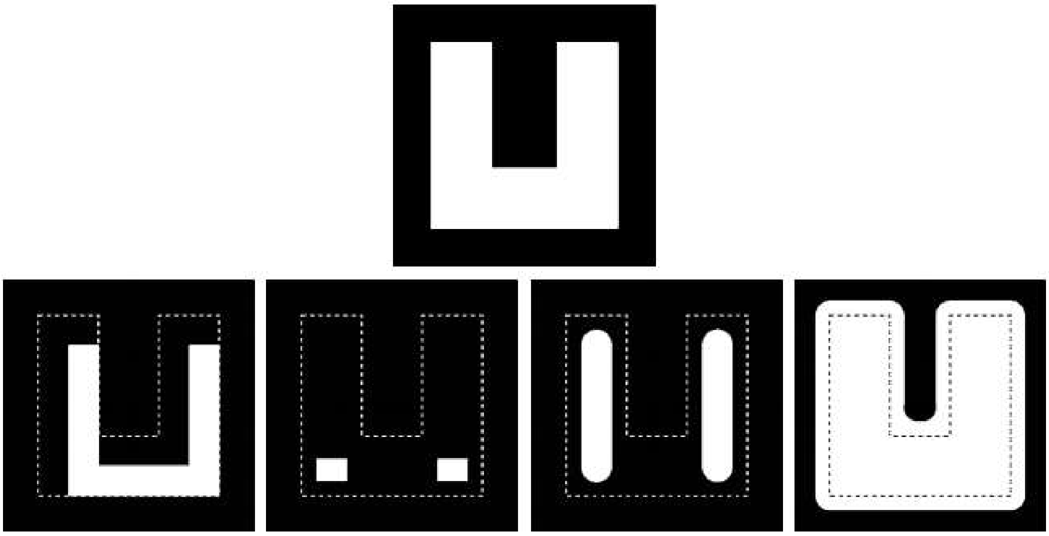
$$A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB) \tag{9.5-15}$$

where $(S_k(A) \oplus kB)$ denotes k successive dilations of $S_k(A)$; that is,

$$(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B) \oplus \ldots) \oplus B) \tag{9.5-16}$$

1. Exercise question 9.5

★**9.5** With reference to the image shown, give the structuring element and morphological operation(s) that produced each of the results shown in images (a) through (d). Show the origin of each structuring element clearly. The dashed lines show the boundary of the original set and are included only for reference. Note that in (d) all corners are rounded.

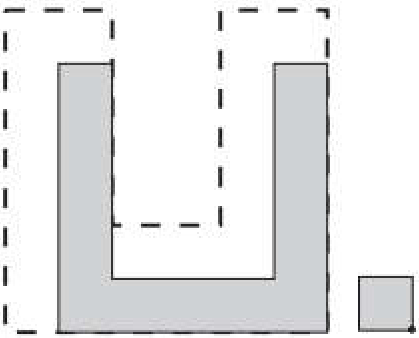


(a) (b) (c) (d)

Solution is –

Refer to Fig. P9.5. The center of each structuring element is shown as a black dot.

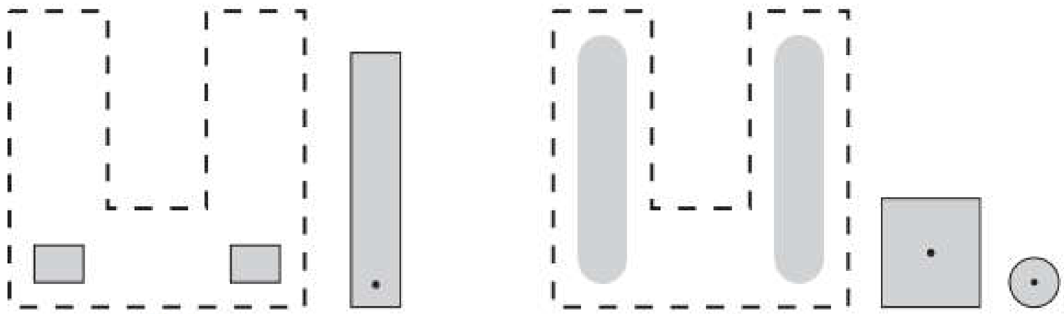
(a) This solution was obtained by eroding the original set (shown dashed) with the structuring element shown (note that the origin is at the bottom, right).



(a)

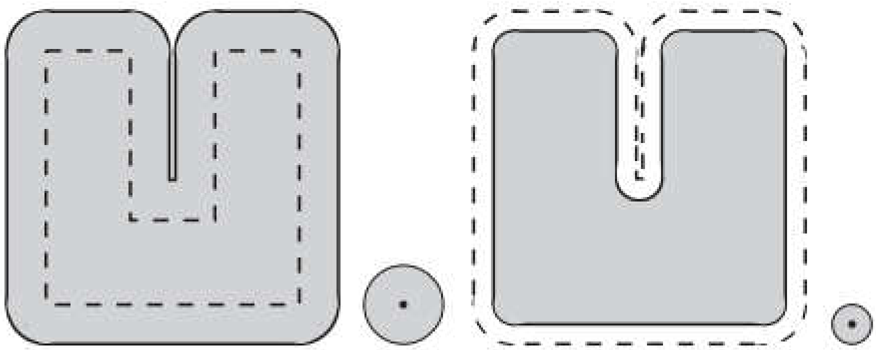
(b) This solution was obtained by eroding the original set with the tall rectangular structuring element shown.

(c) This solution was obtained by first eroding the image shown down to two vertical lines using the rectangular structuring element (note that this elements is slightly taller than the center section of the "U" figure). This result was then dilated with the circular structuring element.



(b) (c)

(d) This solution was obtained by first dilating the original set with the large disk shown. The dilated image was eroded with a disk whose diameter was equal to one-half the diameter of the disk used for dilation.



(d)

2. Exercise question 9.6

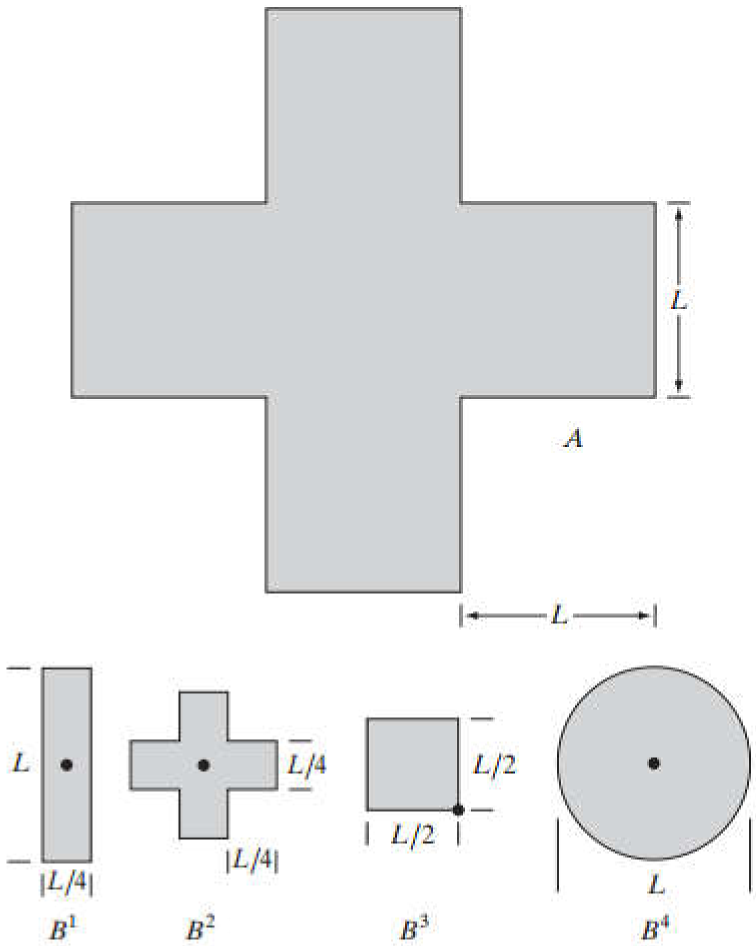
Let A denote the set shown shaded in the following figure. Refer to the structuring elements shown (the black dots denote the origin). Sketch the result of the following morphological operations:

(a) $(A \ominus B^4) \oplus B^2$

(b) $(A \ominus B^1) \oplus B^3$

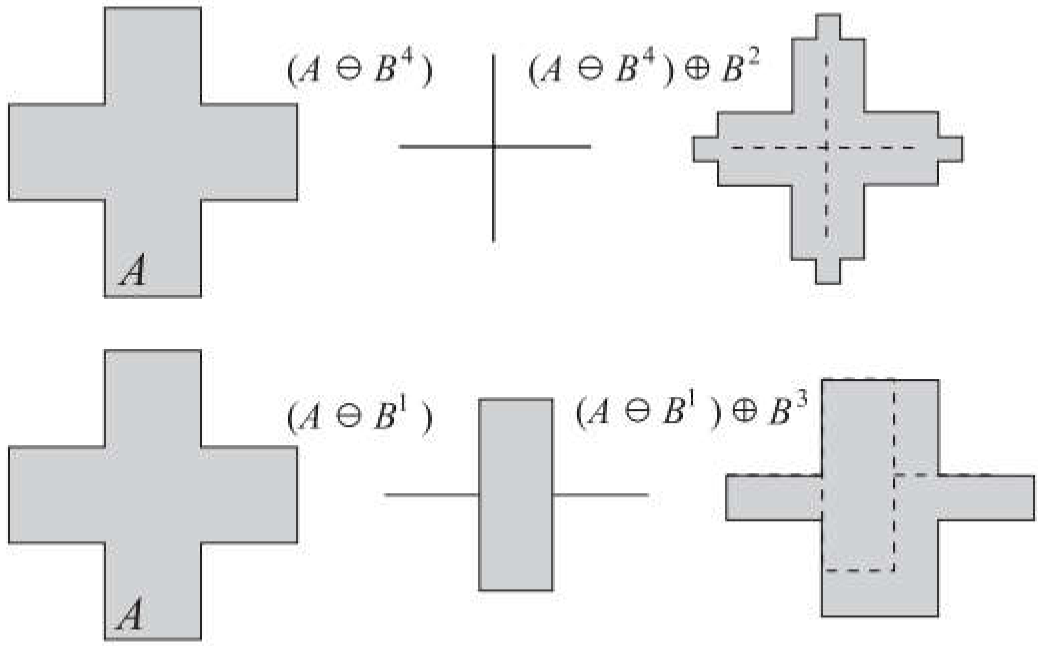
(c) $(A \oplus B^1) \oplus B^3$

(d) $(A \oplus B^3) \ominus B^2$

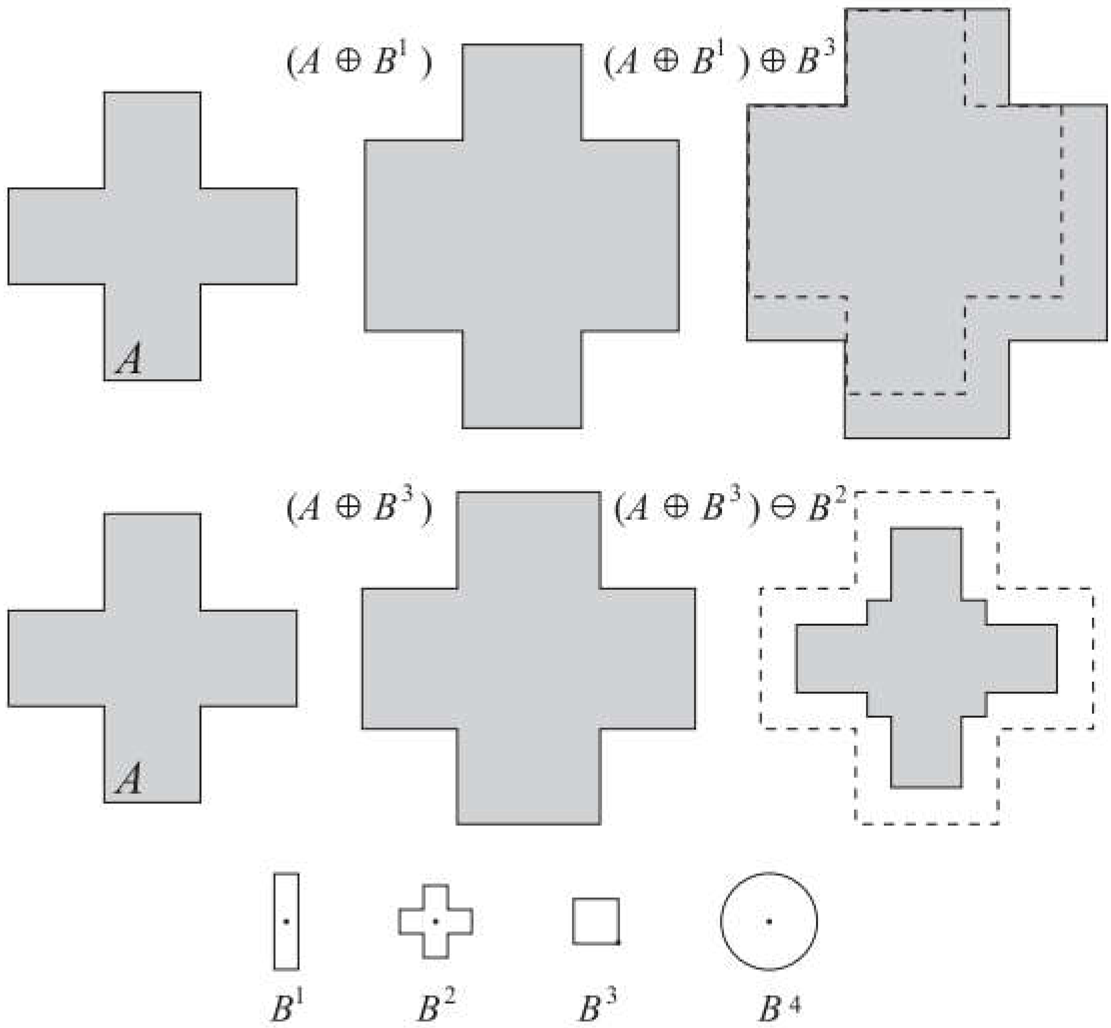


L
A
L
L
L/4
|L/4|
L/2
L/2
L
|L/4|
B^1
B^2
B^3
B^4

Solution-



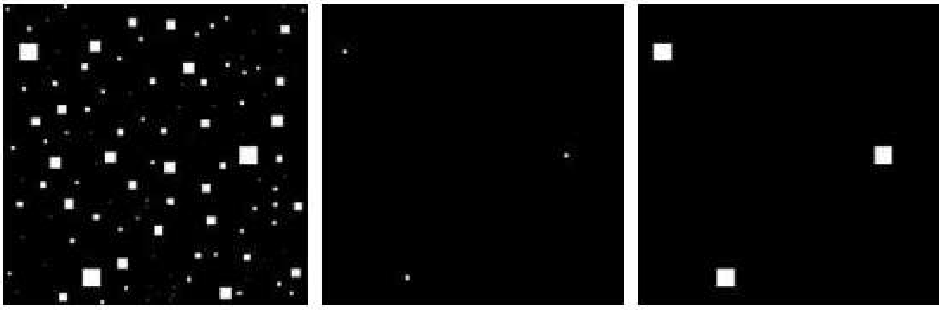
$(A \ominus B^4)$
$(A \ominus B^4) \oplus B^2$
A
$(A \ominus B^1)$
$(A \ominus B^1) \oplus B^3$
A



$(A \oplus B^1)$
$(A \oplus B^1) \oplus B^3$
A
$(A \oplus B^3)$
$(A \oplus B^3) \ominus B^2$
A
B^1
B^2
B^3
B^4

3. Exercise question 9.18

Consider the three binary images shown in the following figure. The image on the left is composed of squares of sizes 1, 3, 5, 7, 9, and 15 pixels on the side. The image in the middle was generated by eroding the image on the left with a square structuring element of 1s, of size 13 × 13 pixels, with the objective of eliminating all the squares, except the largest ones. Finally, the image on the right is the result of dilating the image in the center with the same structuring element, with the objective of restoring the largest squares. You know that erosion followed by dilation is the opening of an image, and you know also that opening generally does not restore objects to their original form. Explain why full reconstruction of the large squares was possible in this case.



Solution –

Problem 9.18

It was possible to reconstruct the three large squares to their original size because they were not completely eroded and the geometry of the objects and structuring element was the same (i.e., they were squares). This also would have been true if the objects and structuring elements were rectangular. However, a complete reconstruction, for instance, by dilating a rectangle that was partially eroded by a circle, would not be possible.