Chapter 2 Image Enhancement in Spatial Domain

Introduction

Image Enhancement is the processing of images to improve their appearance to human viewers, in terms of better contrast and visibility of features of interest, or to enhance their performance in subsequent computer aided analysis and diagnosis. The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application. It is the process that improves the quality of the image for a specific application. It does not add any extra information to the original image but improves the subjective quality of the images by working with the existing data.

Image enhancement approaches fall into two categories: i) spatial domain methods ii) frequency domain methods. The term spatial domain refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image. Frequency domain processing procedures are based on reforming the Fourier transform of an image.

Spatial Domain refers to the image plane itself, the image processing methods in this category are based on direct manipulation of pixels in an image. Two principles categories of spatial processing are Spatial Processing and Spatial Filtering.

In Frequency Domain methods, the image is first transferred in to frequency domain. It means that, the Fourier Transform of the image is computed first which will be discussed in next chapter in detail.

2.1 Spatial Domain Methods/Grey Level Transformations

As indicated previously, the term spatial domain refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression:

$g(x,y) = T[f(x,y)]$

Where f(x,y) in the input image, g(x,y) is the processed image and T is as operator on f, defined over some neighborhood of (x,y) In addition, T can operate on a set of input images.

2.1.1 Point Processing

The simplest kind of range transformations are these independent of position x,y:

g = T(f)

This is called point processing. The simplest form of T, is when the neighborhood of size 1X1 (that is a single pixel). In this case, g depends only on the value of f at (x,y), and T becomes a grey-level (also called intensity or mapping) transformation function of the form: s = T (r) Where, for simplicity in notation, r and s are variables denoting, respectively, the grey level of f(x,y) and g(x,y) at any point (x,y)

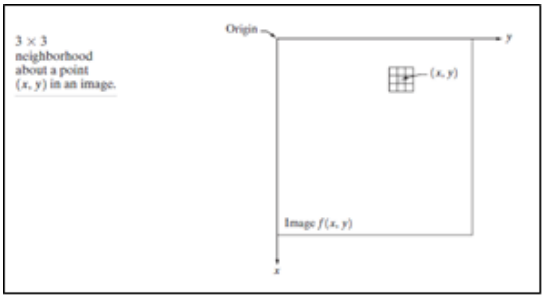


Figure 2.1 3X3 neighborhood about a point (x,y) in an image

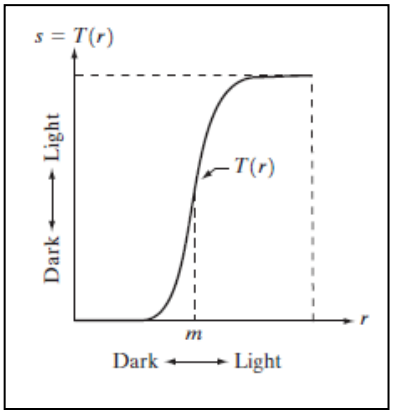
2.1.2 Contrast Stretching

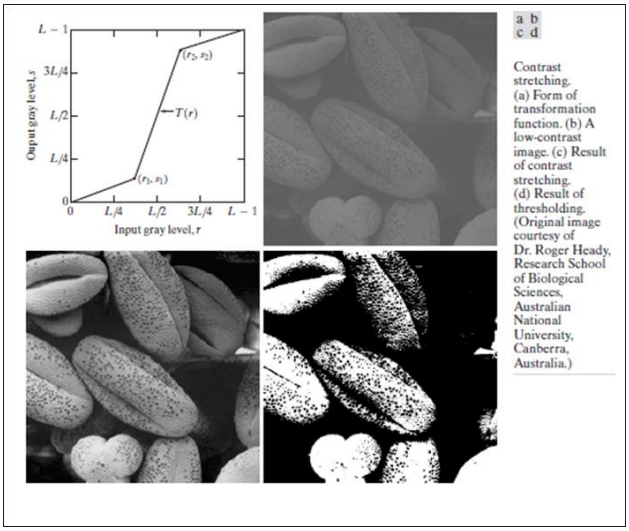
One of the simplest piecewise linear transformation (is type of gray level transformation that is used for image enhancement . It is a spatial domain method. It is used for manipulation of an image so that the result is more suitable than the original for a specific application) is a contrast stretching transformation. Low Contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even setting of lens aperture during image acquisition. **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 medium.

If T(r) has the form as shown in the figure below, the effect of applying the transformation to every pixel of f to generate the corresponding pixels in g would:

Produce higher contrast than the original image, by:

- Darkening the levels below m in the original image
- Brightening the levels above m in the original image





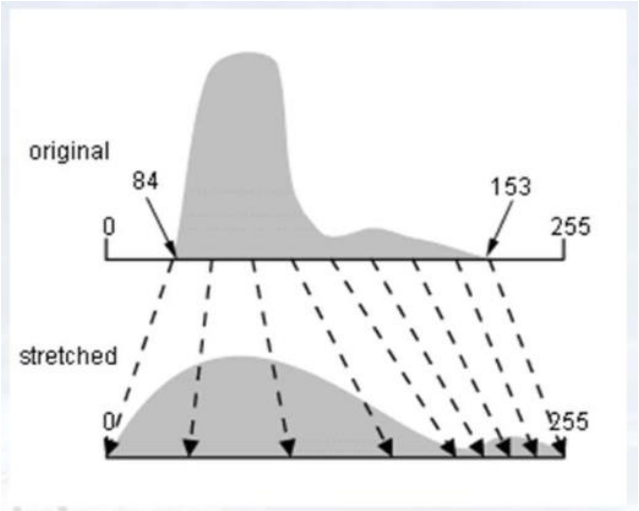
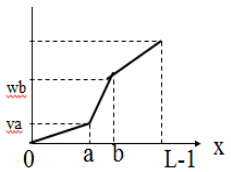


Figure 2.2: Contrast Stretching

- Figure (a) 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 gray levels.
- If r1 = r2, s1 = 0 and s2 = L-1, the transformation becomes a *thresholding function* that creates a binary image.
- Intermediate values of (r1, s1) and (r2, s2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.
- In general, r1 ≤ r2 and s1 ≤ s2 is assumed, so the function is always increasing. Figure (b) shows an 8-bit image with low contrast.
- Fig. (c) shows the result of contrast stretching, obtained by setting (r1, s1) = (r_{min}, 0) and (r2, s2) = (r_{max},L-1) where r_{min} and r_{max} denote the minimum and maximum gray levels in the image, respectively. Thus, the transformation function stretched the levels linearly from their original range to the full range [0, L-1].

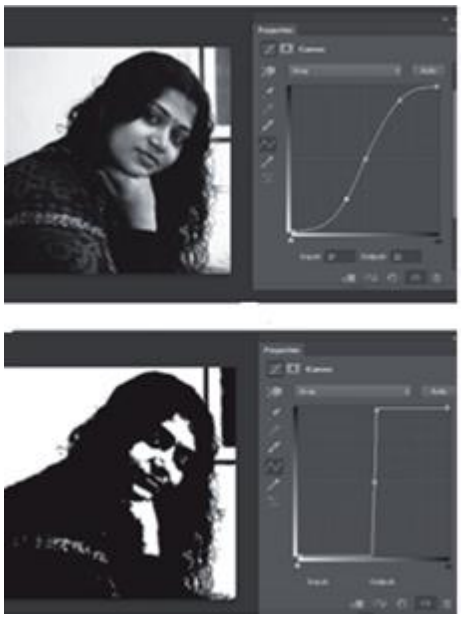
$$s = \begin{cases} lr & 0 \le r < a \\ m(r-a)+v_a & a \le r < b \\ n(r-b)+w_b & b \le r < L-1 \end{cases}$$



- Finally, Fig.(d) shows the result of using the

thresholding function defined previously, with r1=r2=m, the mean gray level in the image.

2.1.3 Thresholding



Thresholding is the simplest method of image segmentation.

From a grayscale image, thresholding can be used to create binary images. The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity $I_{i,j}$ is less than some fixed constant T (that is $I_{i,j}$), or a white pixel if the image intensity is greater than that constant.

Expression

s = 0 if r <= m

S = L – 1 if r > m

Figure 2.3 Binary Thresholding

In Figure 2.3(b) where the higher intensities (r>128) are mapped to the highest intensity (s=255) and lower intensities (r<128) are mapped to the lowest intensity (s=0), the transformation takes the form of thresholding for binarization, as the output image would have only two intensities (binary image)— white and black.

Problem

Find the thresholding of given 8 bit image .Let's assume r<=128 →s=0 r>128→s=255

Original Image

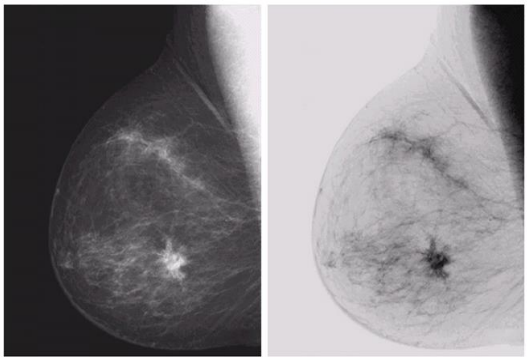
121	205	217
139	127	157
252	117	236

Thresholding

0	255	255
255	0	255
255	0	255

2.1.4 Digital Negative

The negative of an image with gray level in the range [0, L-1], where L = Largest value in an image, is obtained by using the negative transformation's expression:**s = L – 1 – r** Which reverses the intensity levels of an input image, in this manner produces the equivalent of a photographic negative. The negative transformation is suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area are dominant in size



a b

(a) Original digital mammogram. (b) Negative image obtained using the negative transformation

Figure 2.4 Digital Negative/Image Negative/Negative Transformations

Problem

Find the digital negative of given 8 bit image

121	205	217
139	127	157
252	117	236

Original Image

134	50	33
116	128	98
3	138	19

Digital Negative of Original Image

The digital negative of an image can be given as, s = (L-1)-r

Where s is output pixel value, r is input pixel value and L-1 is maximum pixel value.

For 8 bits per pixel image maximum pixel value is 255.

2.1.5 Intensity Level Slicing/Gray-level Slicing

This technique is used to highlight a specific range of gray levels in a given image. It can be implemented in several ways, but the two basic themes are:

- One approach is to display a high value for all gray Levels in the range of interest and a low value for all other gray levels. This transformation, shown in Figure 2.5(a), produces a binary image.
- The second approach, based on the transformation shown in Figure 2.5 (b), brightens the desired range of gray levels but preserves gray levels unchanged. Figure 2.5 (c) shows a gray scale image and Figure 2.5 (d) shows the result of using the transformation in Figure 2.5 (a).

Expression:

- Without Back ground:

  s = L – 1, if A <= r <= B

  = 0 Otherwise
- With Back ground:

  s = L – 1, if A <= r <= B

  = r Otherwise

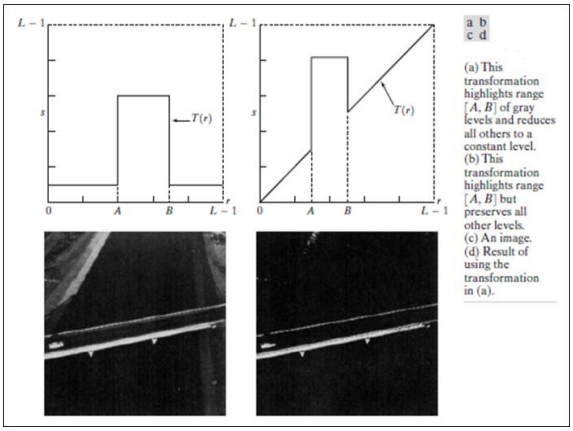


Figure 2.5 Intensity Level Slicing

Problem

For the 3-bit 4*4 size image perform Intensity Level Slicing with background and without back ground.

1	2	3	0
2	4	6	7
5	2	4	3
3	2	6	1

Solution:

Since image is 3-bit image .Image 2^3=8 Levels than L-1=8-1=7

Let’s assume for without background

3<=f(x,y)<=5→7

Otherwise----→0

(a) Without Background

0	0	7	0
7	7	0	0
7	0	7	7
7	0	0	0

Let's assume for with background

3<=f(x,y)<=5→7

Otherwise----→r

(b) With Background

1	2	7	0
2	7	6	7
7	2	7	7
7	2	6	1

2.1.6 Bit Plane Slicing

Pixels are digital numbers, each one composed of bits. Instead of highlighting gray-level range, we could highlight the contribution made by each bit. This method is useful and used in image compression. Most significant bits contain the majority of visually significant data .To highlight the contribution made to the total image appearance by specific bits. i.e. Assuming that each pixel is represented by 8 bits, the image is composed of 8 1-bit planes. Plane 0 contains the least significant bit and plane 7 contains the most significant bit. Only the higher order bits (top four) contain visually significant data. The other bit planes contribute the more subtle details.

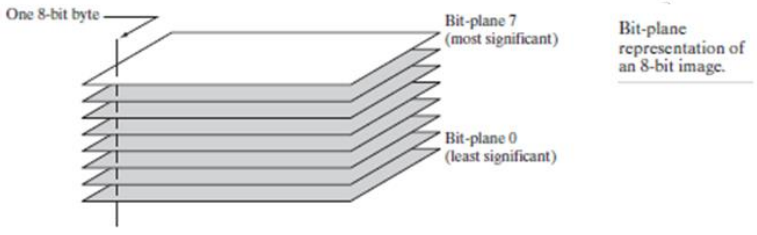


Figure 2.6 Bit Plane Slicing

- Example:

  3 × 3 image

1	2	0
4	3	2
7	5	2

Binary Image

001	010	000
100	011	010
111	101	010

- Example:

  LSB image

1	0	0
0	1	0
1	1	0

Less contribution in Image

HSB Image

0	0	0
1	0	0
1	1	0

High Contribution in Image

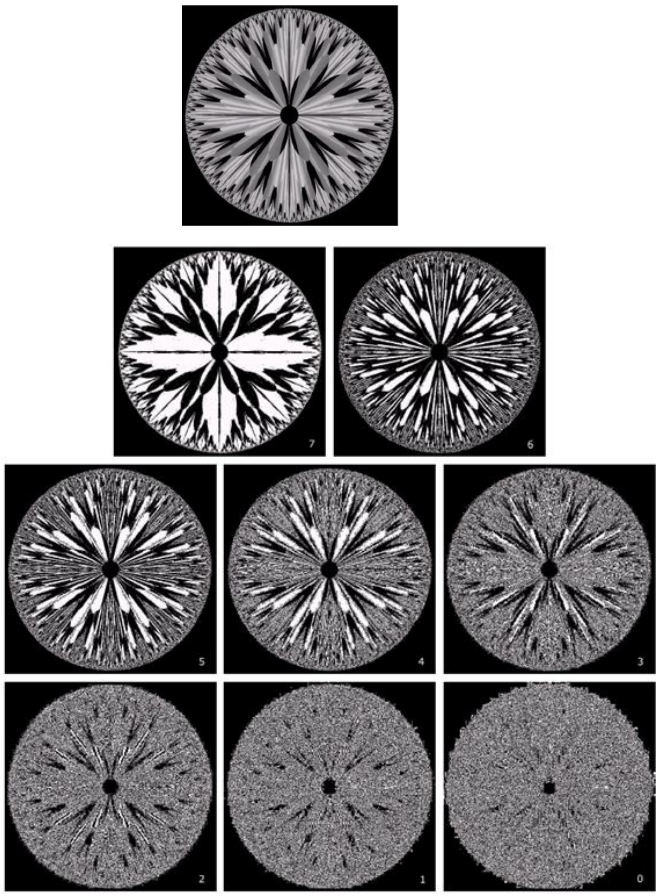


Figure 2.7(a) An 8-Bit fractal image (b) 8 bit planes of image. The number at the bottom right of each image identifies bit plane

Other Spatial Domain Methods/Grey Level Transformations

(a)Logarithmic Transformations

Logarithmic transformation further contains two type of transformation. Log transformation and inverse log transformation.

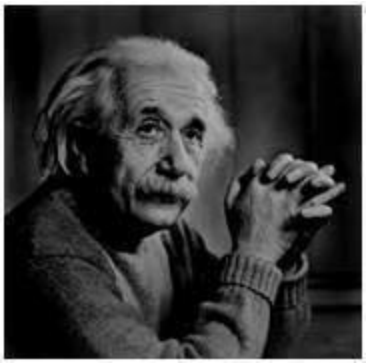
Log transformation

The log transformations can be defined by this formula

$s = c \log(r + 1)$.

Where s and r are the pixel values of the output and the input image and c is a constant. The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity. So 1 is added, to make the minimum value at least 1.During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values. The higher pixel values are kind of compressed in log transformation. This result in following image enhancement. The value of c in the log transform adjusts the kind of enhancement we are looking for.

Input Image



Log Transform Image

The inverse log transform is opposite to log transform.

(b)Power – Law transformations

There are further two transformation is power law transformations, that include nth power and nth root transformation. These transformations can be given by the expression:

$s=cr^{\gamma}$

This symbol γ is called gamma, due to which this transformation is also known as gamma transformation.

Variation in the value of γ varies the enhancement of the images. Different display devices / monitors have their own gamma correction, that's why they display their image at different intensity.

This type of transformation is used for enhancing images for different type of display devices. The gamma of different display devices is different. For example Gamma of CRT lies in between of 1.8 to 2.5 that means the image displayed on CRT is dark.

Correcting gamma .

$s=cr^{\gamma}$

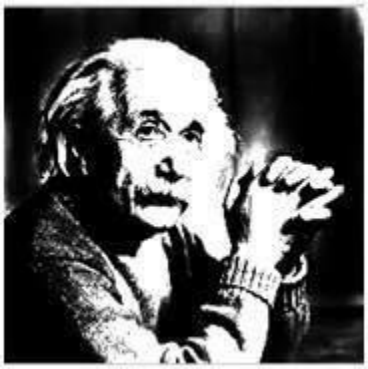
$s=cr^{(1/2.5)}$

The same image but with different gamma values has been shown here.

For example

Gamma = 10

Gamma = 8



Gamma = 6

2.2 Histogram Processing and Equalization

Histogram:

- In Statistics, Histogram is a graphical representation showing a visual impression of the distribution of data.
- An Image Histogram is a type of histogram that acts as a graphical representation of the lightness/color distribution in a digital image. It plots the number of pixels for each value.
- The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function $h(r_k) = n_k$, where r_k is the kth gray level and n_k is the number of pixels in the image having gray level r_k.
- It is common practice to normalize a histogram by dividing each of its values by the total number of pixels in the image, denoted by n. Thus, a normalized histogram is given by $p(r_k) = n_k / n$, for k = 0, 1, …, L -1.
- Thus, $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k. Note that the sum of all components of a normalized histogram is equal to 1.

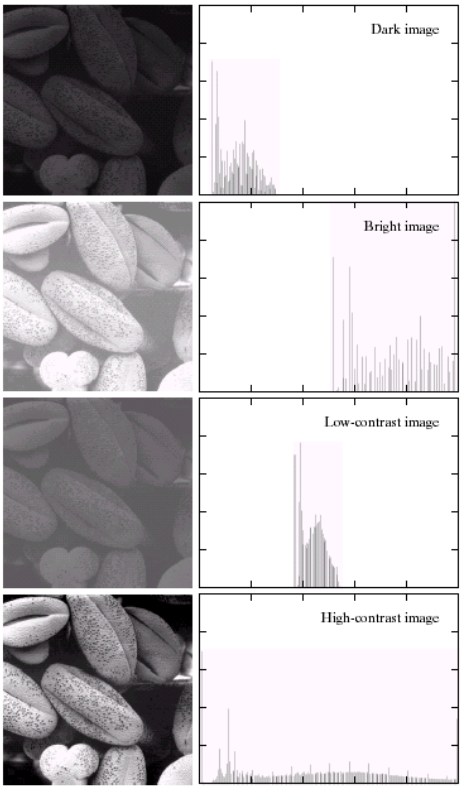


Figure 2.8 Different Contrast images.

- As an introduction to the role of histogram processing in image enhancement, consider Figure 2.8 shown in four basic gray-level characteristics: dark, light, low contrast, and high contrast. The right side of the figure shows the histograms corresponding to these images.
- The horizontal axis of each histogram plot corresponds to gray level values, r_k.
- The vertical axis corresponds to values of $h(r_k)=n_k$ or $p(r_k)=n_k/n$ if the values are normalized.
- Thus, as indicated previously, these histogram plots are simply plots of $h(r_k)=n_k$ versus r_k or $p(r_k)=n_k/n$ versus r_k.

Why Histogram?

- Histograms are the basis for numerous spatial domain processing techniques
- Histogram manipulation can be used effectively for image enhancement
- Histograms can be used to provide useful image statistics. Information derived from histograms are quite useful in other image processing applications, such as image compression and segmentation

What is the histogram of a digital image?

The histogram of a digital image with gray values $r_0,r_1,\ldots,r_{L-1}$ is the discrete function

$$p(r_k) = \frac{n_k}{n}$$

n_k: Number of pixels with gray value r_k

n: total Number of pixels in the image

The function $p(r_k)$ represents the fraction of the total number of pixels with gray value r_k.

Histogram Equalization

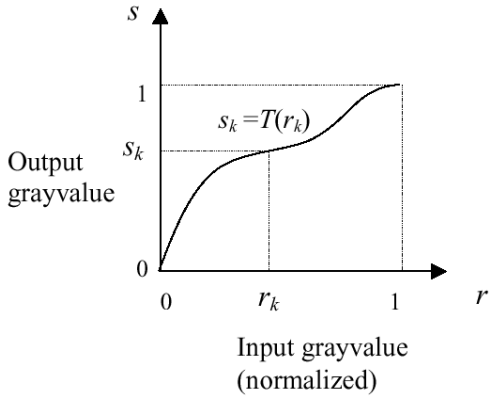
The histogram equalization is an approach to enhance a given image. The approach is to design a transformation $T(.)$ such that the gray values in the output is uniformly distributed in [0, 1].Let us assume for the moment that the input image to be enhanced has continuous gray values, with $r = 0$ representing black and $r = 1$ representing white.

We need to design a gray value transformation $s = T(r)$, based on the histogram of the input image, which will enhance the image. The objective of histogram equalization is not only to spread the dynamic range but also to have equal pixels in all the gray levels· As before, we assume that:

(1) $T(r)$ is a monotonically increasing function for $0 \leq r \leq 1$ (preserves order from black to white).

(2) $T(r)$ maps [0,1] into [0,1] (preserves the range of allowed Gray values).

Let us denote the inverse transformation by $r = T^{-1}(s)$. We assume that the inverse transformation also satisfies the above two conditions.



We consider the gray values in the input image and output image as random variables in the interval [0, 1]. · Let $p_{in}(r)$ and $p_{out}(s)$ denote the probability density of the Gray values in the input and output images. ·If $p_{in}(r)$ and T(r) are known, and $r = T^{-1}(s)$ satisfies condition 1, we can write (result from probability theory):

$$p_{out}(s) = \left[p_{in}(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)}$$

One way to enhance the image is to design a transformation $T(.)$ such that the gray values in the output is uniformly distributed in [0, 1], i.e. $p_{out}(s) = 1, \quad 0 \leq s \leq 1$.

In terms of histograms, the output image will have all gray values in "equal proportion".

This technique is called **histogram equalizatio**n.

Next we derive the gray values in the output is uniformly distributed in [0, 1]. Consider the transformation

$$s = T(r) = \int_0^r p_{in}(w)dw, \qquad 0 \le r \le 1$$

From the previous equation and using the fundamental theorem of calculus, Note that this is the cumulative distribution function (CDF) of p_{in} *(r)* and satisfies the previous two conditions. · From the previous equation and using the fundamental theorem of calculus,

$$\frac{ds}{dr} = p_{in}(r)$$

Therefore, the output histogram is given by

$$p_{out}(s) = \left[p_{in}(r) \cdot \frac{1}{p_{in}(r)} \right]_{r=T^{-1}(s)} = [1]_{r=T^{-1}(s)} = 1, \qquad 0 \le s \le 1$$

The output probability density function is uniform, regardless of the input. Thus, using a transformation function equal to the CDF of input gray values r, we can obtain an image with uniform gray values. This usually results in an enhanced image, with an increase in the dynamic range of pixel values.

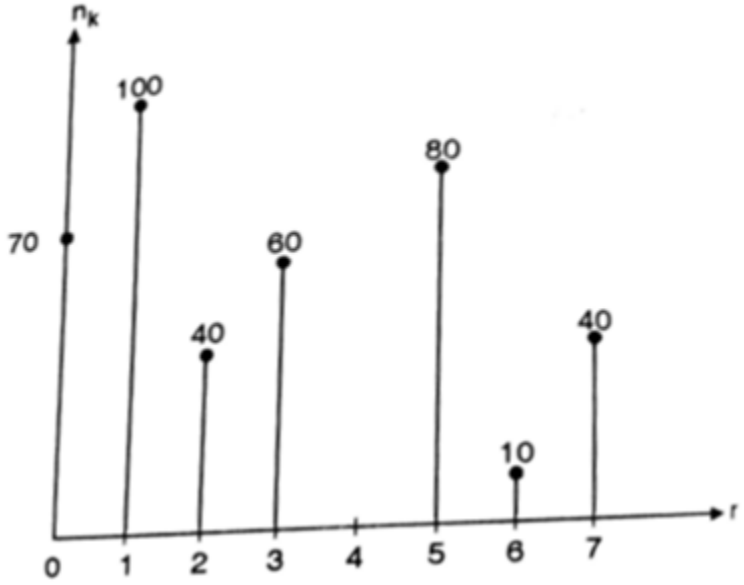
Problem

Equalize the given histogram

Original Image

Gray Level	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
No. of. Pixel	**70**	**100**	**40**	**60**	**0**	**80**	**10**	**40**

Step 1: Plot the original Histogram



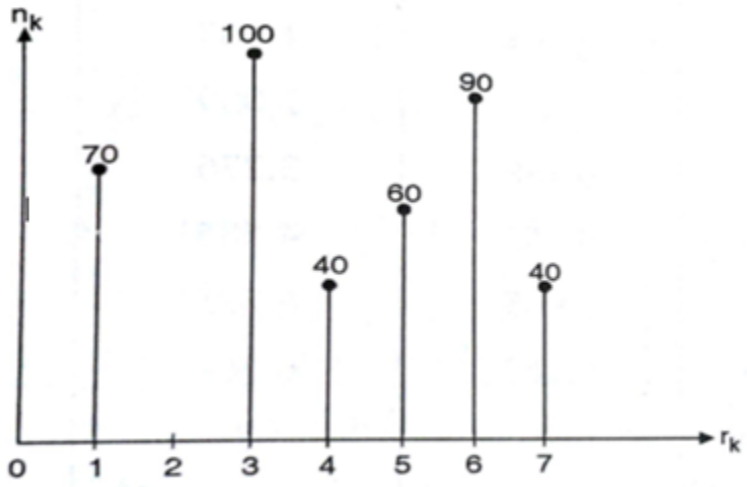
Step 2 : Perform Histogram Equalization

Grey Level	n_k	PDF=n_k/n	CDF=$\sum PDF$	CDF x (L-1) CDF x 7	Rounding off (New Grey Level)
0	70	0.175	0.175	1.22	1
1	100	0.25	0.425	2.97	3
2	40	0.1	0.525	3.67	4
3	60	0.15	0.675	4.72	5
4	0	0	0.675	4.72	5
5	80	0.2	0.875	6.12	6
6	10	0.025	0.9	6.3	6
7	40	0.1	1	7	7
Total	n=400				

Step 3: Make new histogram table

Old Grey Level	n_k	New Grey Level	New n_k
0	70	1	70
1	100	3	100
2	40	4	40
3	60	5	60 + 0 = 60
4	0	5	
5	80	6	80 + 10 = 90
6	10	6	
7	40	7	40

Step 4: Plot the equalized Histogram



2.3 Enhancement Using Arithmetic and Logic Operations

Two images of the same size can be combined using operations of addition, subtraction, multiplication, division, logical AND, OR, XOR and NOT. Such operations are done on pairs of their corresponding pixels. Often only one of the images is a real picture while the other is a machine generated mask. The mask often is a binary image consisting only of pixel values 0 and 1.

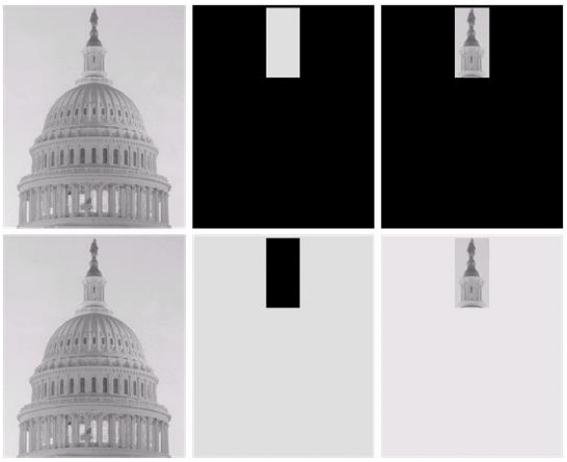
Arithmetic Operations

Addition, Subtraction, Multiplication, and Division

Logic Operations

AND, OR, NOT

AND and OR operations are masking-selecting sub image in an image, Subtraction and Additions are used for image enhancement



a b c
d e f

(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).

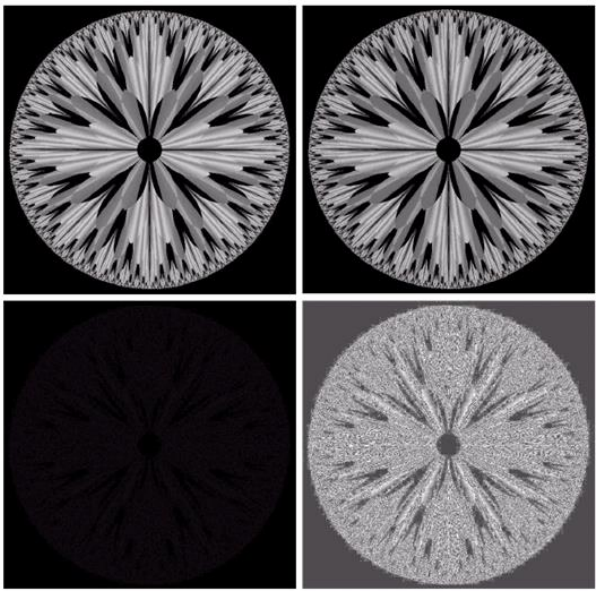
Figure 2.9 Enhancement Using Arithmetic and Logic Operations

g(x, y) = f(x, y)–h(x, y)

- Higher-order bit plane – visual relevant detail
- Lower-order bit plane – fine detail or imperceptible detail

a b
c d

(a) Original fractal image. (b) Result of setting the four lower-order bit planes to zero. (c) Difference between (a) and (b). (d) Histogram-equalized difference image. (Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).



Original

Noise with zero mean

$$g(x,y) = f(x,y) + \eta(x,y)$$

$$\bar{g}(x,y) = \frac{1}{k}\sum_{i=1}^{k} g_i(x,y)$$

$$E\left[\bar{g}(x,y)\right] = f(x,y)$$

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{k}\sigma^2_{\eta(x,y)}$$

Figure 2.10 Enhancement Using Arithmetic and Logic Operations

2.4 Basics of Spatial Filters

Two types of Spatial Filters

(a) Point Processing

$$g(x,y) = T\,[\,f(x,y)\,]$$

(b) Mask Processing Method

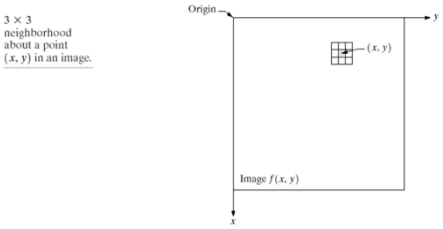


Figure 2.11 Point Processing

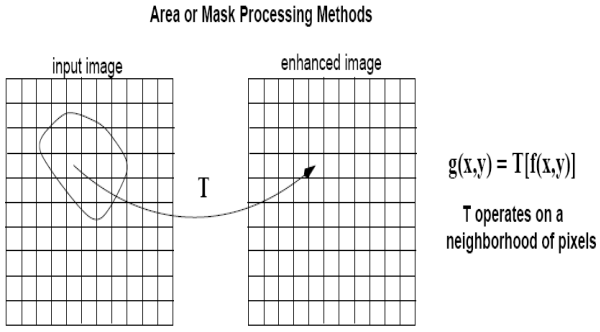
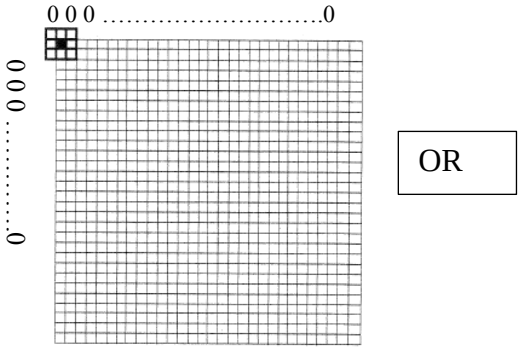
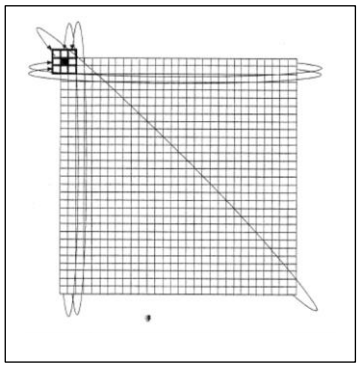


Figure 2.12 Mask Processing

Handling Pixels Close to Boundaries



OR



Spatial filtering is defined by:

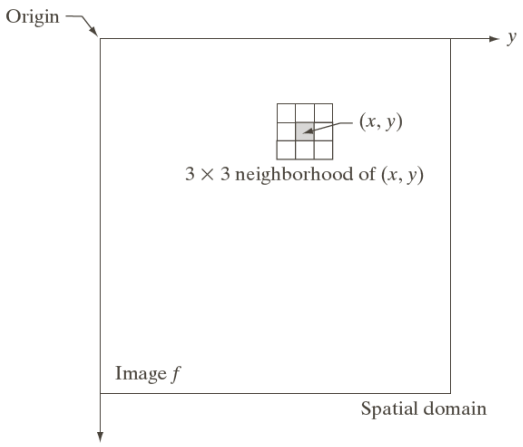
(1) A neighborhood

(2) An operation that is performed on the pixels inside the neighborhood

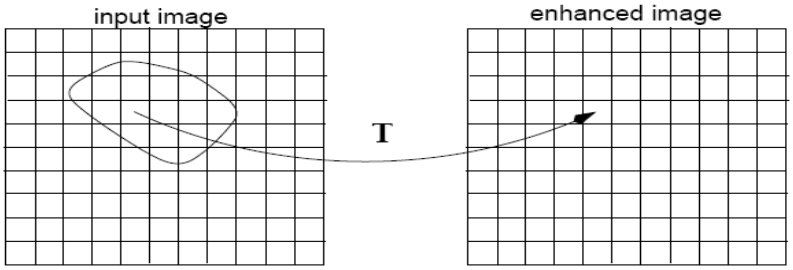
- Typically, the neighborhood is rectangular and its size

is much smaller than that of *f(x,y)*

- e.g., 3x3 or 5x5



Area or Mask Processing Methods



g(x,y) = T[f(x,y)]

T operates on a neighborhood of pixels

$z5' = R = w1z1 + w2z2 + \ldots + z9w9$

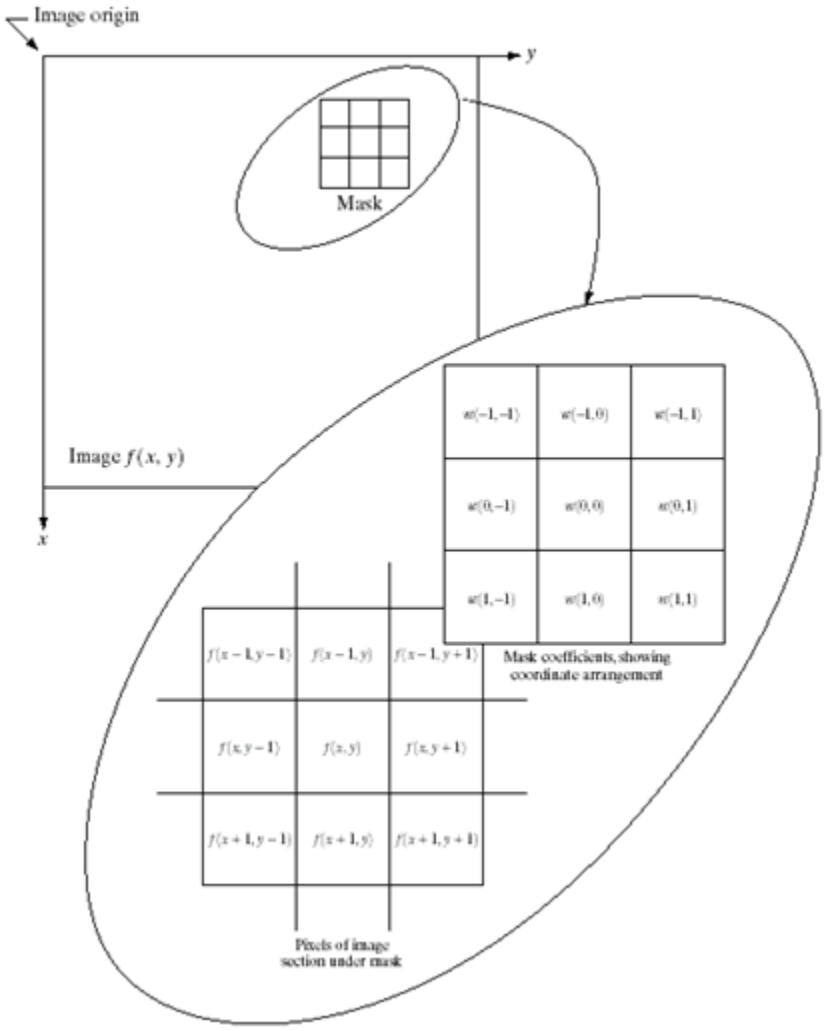
- A filtered image is generated as the center of the mask moves to every pixel in the input image.

Linear vs Non-Linear Spatial Filtering Methods

- A filtering method is linear when the output is a weighted sum of the input pixels. Eg. Mean filter

$z5' = R = w1z1 + w2z2 + \ldots + z9w9$

- Methods that do not satisfy the above property are called non-linear.
  - e.g., median filter



mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

Figure 13 Spatial Filtering

- Spatial filtering are filtering operations performed on the pixel intensities of an image and not on the frequency components of the image. Use of spatial masks for image processing (spatial filters)

  Linear and nonlinear filters
- Low-pass filters eliminate or attenuate high frequency components in the frequency domain (sharp image details), and result in image blurring.

Response, R, of an $m \times n$ mask at any point (x, y)

$$R = \sum_{i=1}^{mn} w_i z_i$$

Another representation of a general 3 × 3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Special consideration is given when the center of the filter approach the boarder of the image.

2.5 Smoothening and Sharping Spatial Filters

- The word "filtering" has been borrowed from the frequency domain.
- Filters are classified as:
  - Low-pass (i.e., preserve low frequencies)
  - High-pass (i.e., preserve high frequencies)
  - Band-pass (i.e., preserve frequencies within a band)

Smoothing filters are used

- Noise reduction
- Smoothing of false contours
- Reduction of irrelevant detail

(a)Low-pass (Averaging) filters

- An image is smoothed by decreasing the disparity between pixel values by averaging nearby pixels.
- It eliminate or attenuate high frequency components in the frequency domain (sharp image details), and result in image blurring i.e.it preserves low frequencies.
- Undesirable side effect of smoothing filters
  - Blur edges
- Weighted average filter reduces blurring in the smoothing process.
- The low-pass filters usually employ moving window operator which affects one pixel of the image at a time, changing its value by some function of a local region (window) of pixels. The operator moves over the image to affect all the pixels in the image.

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

1	1	1
1	1	1
1	1	1

Figure 14 Box Filter

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

1	2	1
2	4	2
1	2	1

Figure 15 Weighted Average

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

Weighted average filter reduces blurring in the smoothing process.

Figure 16: Spatial filtering

Example of Low Pass Filter

4	3	2	1
3	1	2	4
5	1	6	2
2	3	5	6

*

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Original Image * Convolution Mask

Now we can perform zero padding or pixel replication to handle boundary pixels. In this problem we are considering pixel replication.

Pixel Replication

4	4	3	2	1	1
4	4	3	2	1	1
3	3	1	2	4	4
5	5	1	6	2	2
2	2	3	5	6	6
2	2	3	5	6	6

4	4	3	2	1	1
4	4	3	2	1	1
3	3	1	2	4	4
5	5	1	6	2	2
2	2	3	5	6	6
2	2	3	5	6	6

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

$$f(x_1y_1) = 4\times\frac{1}{9} + 4\times\frac{1}{9} + 3\times\frac{1}{9} + 4\times\frac{1}{9} + 4\times\frac{1}{9} + 3\times\frac{1}{9} + 3\times\frac{1}{9} + 3\times\frac{1}{9} + 1\times\frac{1}{9} = 3.2$$

4	4	3	2	1	1
4	4	3	2	1	1
3	3	1	2	4	4
5	5	1	6	2	2
2	2	3	5	6	6
2	2	3	5	6	6

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

$$f(x_1y_1) = 4\times\frac{1}{9} + 4\times\frac{1}{9} + 3\times\frac{1}{9} + 4\times\frac{1}{9} + 4\times\frac{1}{9} + 3\times\frac{1}{9} + 3\times\frac{1}{9} + 3\times\frac{1}{9} + 1\times\frac{1}{9} = 3.2$$

$$f(x_2y_2) = 4\times\frac{1}{9} + 3\times\frac{1}{9} + 2\times\frac{1}{9} + 4\times\frac{1}{9} + 3\times\frac{1}{9} + 2\times\frac{1}{9} + 3\times\frac{1}{9} + 1\times\frac{1}{9} + 2\times\frac{1}{9} = 2.9$$

We slide the window by one column till we scan the whole image , each centered pixel is replaced with the current value.

4	4	3	2	1	1
4	3.2	2.9	2.1	2	1
3	3.2	3	2.4	2.6	4
5	2.7	3.1	3.3	4.1	2
2	2.7	3.5	4.1	4.8	6
2	2	3	5	6	6

=

4	4	3	2	1	1
4	3	3	2	2	1
3	3	3	2	3	4
5	3	3	3	4	2
2	3	4	4	5	6
2	2	3	5	6	6

This is the final result of Low Pass Filtering.

Order-statistics 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.

3 × 3 Median filter [10 125 125 135 141 141 144 230 240] = 141

3 × 3 Max filter [10 125 125 135 141 141 144 230 240] = 240

3 × 3 Min filter [10 125 125 135 141 141 144 230 240] = 10

Median filter

- Used primarily for noise reduction (eliminates isolated spikes)
- The gray level of each pixel is replaced by the median of the gray levels in the neighborhood of that pixel (instead of by the average as before).
- The averaging filter removes the noise by blurring the edges.
- When we need to eliminate salt and pepper noise, median filter is effective.

  The steps to perform median Filtering:
- Assume a 3 × 3 empty mask
- Place the empty mask at the left hand corner
- Arrange the 9 pixels in ascending or descending order.
- Choose the median value from these nine values.
- Place the median at the center
- Move the mask in a similar fashion to the averaging filter.
- Poor performance when the number of noise pixels in the window is greater than half the number of pixels in the window

- eliminates isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2/2$.
    - Used primarily for noise reduction (eliminates isolated spikes)
    - The gray level of each pixel is replaced by the median of the gray levels in the neighborhood of that pixel (instead of by the average as before).

Image averaging/Mean Filter is obtained by finding the average of K images.

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^{K} g_i(x, y)$$

It is applied in de-noising the images. A noisy image is defined by:

$$g(x, y) = f(x, y) + \eta(x, y)$$

It is used in removal of blurring in an image and to provide smooth and sharping edges also used for noise reduction

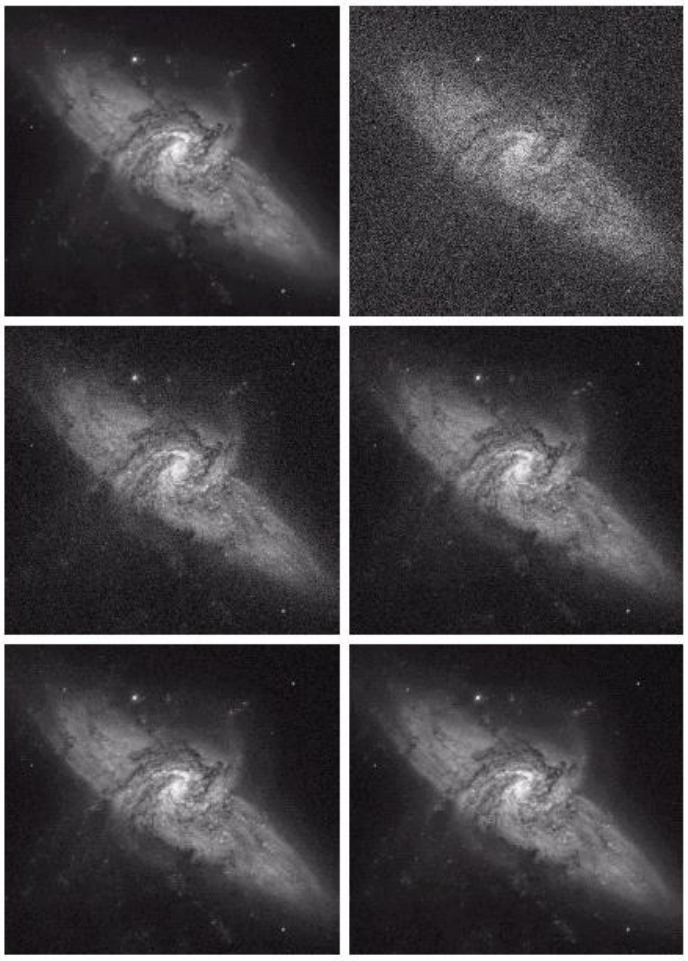


Figure 17 Result of averaging when K=8,16,64,128 of noisy images

High Pass Filter:

- It eliminates the low frequency regions while retaining or enhancing the high frequency components.
- High pass filters are used to sharpen blurred images.
- The masking process is same as low pass filter but mask coefficient is changed. One example of 3 × 3 high pass mask is

-1	**-1**	**-1**
-1	**8**	**-1**
-1	**-1**	**-1**

- The important thing to note is that the sum of the coefficients of the high pass mask has to be equal to zero.

- This is because when we place this mask over the low frequency regions, the result must be zero.

Two issues:

1. Negative value may occur which need to be removed. So replace all negative values by 0 i.e. Dark.
2. The values of the original image at the edge are very large due to center weight 8. To eliminate this problem, we use the mask with scaling function.

Some of the other high pass masks are

High pass image is also obtained by subtracting the low pass image by original image.

0	-1	0
-1	4	-1
0	-1	0

-1	-2	-1
-2	12	-2
-1	-2	-1

$F_{HP}(x,y) = F_{original}(x,y) - F_{LP}(x,y)$

MAGNIFICATION AND INTERPOLATION (IMAGE 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.

- Two different methods of Image Zooming
  (a) Replication (b)Linear Interpolation

(a)Zooming by pixel replication:

We simply replicate each pixel and then replicate each row

Consider the image

Input Image

1	2	3	4
5	6	7	8
9	8	6	7
0	1	2	3

------>

4X4

Image is zoomed into 8X8 image

Output Image by Replication

1	1	2	2	3	3	4	4
1	1	2	2	3	3	4	4
5	5	6	6	7	7	8	8
5	5	6	6	7	7	8	8
9	9	8	8	6	6	7	7
9	9	8	8	6	6	7	7
0	0	1	1	2	2	3	3
0	0	1	1	2	2	3	3

(b)Zooming by Linear Interpolation

In this method instead of replicating each pixel, average of two adjacent pixels along the rows is taken and placed between two pixels.

1	0	2	0	3	0	4	0
0	0	0	0	0	0	0	0
5	0	6	0	7	0	8	0
0	0	0	0	0	0	0	0
9	0	8	0	6	0	7	0
0	0	0	0	0	0	0	0
0	0	1	0	2	0	3	0
0	0	0	0	0	0	0	0

$H =$

1	1.5	2	2.5	3	3.5	4	2
3	3.5	4	4.5	5	5.5	6	3
5	5.5	6	6.5	7	7.5	8	4
7	7	7	6.25	6.5	7	7.5	3.75
9	8.5	8	7	6	6.5	7	3.5
4.5	4.5	4.5	4.25	4	4.5	5	2.5
0	0.5	1	1.5	2	2.5	3	1.5
0	0.25	0.5	0.75	1	1.25	1.5	0.75

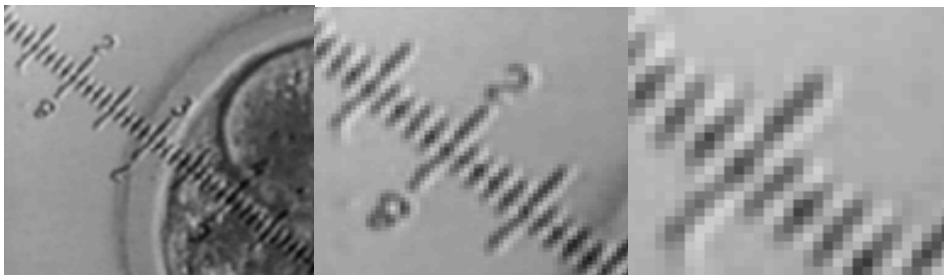
The resulting image is obtained as:

$$v(m,n) = u(k,l)$$

with

$$k = Int[\frac{m}{2}],\ l = Int[\frac{n}{2}]$$

m,n =0, 1, 2,...



$$v_i(m,2n) = u(m,n), \quad o \le m \le M-1, \quad o \le n \le N-1$$

$$v_i(m,2n+1) = \frac{[u(m,n) + u(m,n+1)]}{2}, \quad 0 \le m < M-1$$

$$v(2m,n) = v_i(m,n)$$

$$v(2m+1,n) = \frac{[v_i(m,n) + v_i(m+1,n)]}{2}, \; 0 \le m \le M-1, \; 0 \le n \le 2N-1$$

$$H = \begin{vmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{vmatrix}$$

$$\begin{bmatrix} 1 & 7 \\ 3 & 1 \end{bmatrix} \rightarrow \boxed{\text{Zeros interlacing}} \rightarrow \begin{bmatrix} 1 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \boxed{\text{Rows interpolation}} \rightarrow \begin{bmatrix} 1 & 4 & 7 & 3{,}5 \\ 0 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0{,}5 \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \boxed{\text{Columns interpolation}} \rightarrow \begin{bmatrix} 1 & 4 & 7 & 3{,}5 \\ 2 & 3 & 4 & 2 \\ 3 & 2 & 1 & 0{,}5 \\ 1{,}5 & 1 & 0{,}5 & 0{,}25 \end{bmatrix}$$