Image Enhancement in the Spatial Domain

Principle Objective of Enhancement

- Enhancement refers to accentuation, or sharpening, of image features such as edges, boundaries, or contrast to make graphic display more useful for display and analysis.
- Does not increase the inherent information content in the data, only increase the dynamic range of chosen feature.

Principle Objective of Enhancement

- Process an image so that the result will be more suitable than the original image for a specific application.
- The suitableness is up to each application.
- A method which is quite useful for enhancing an image may not necessarily be the best approach for enhancing another images

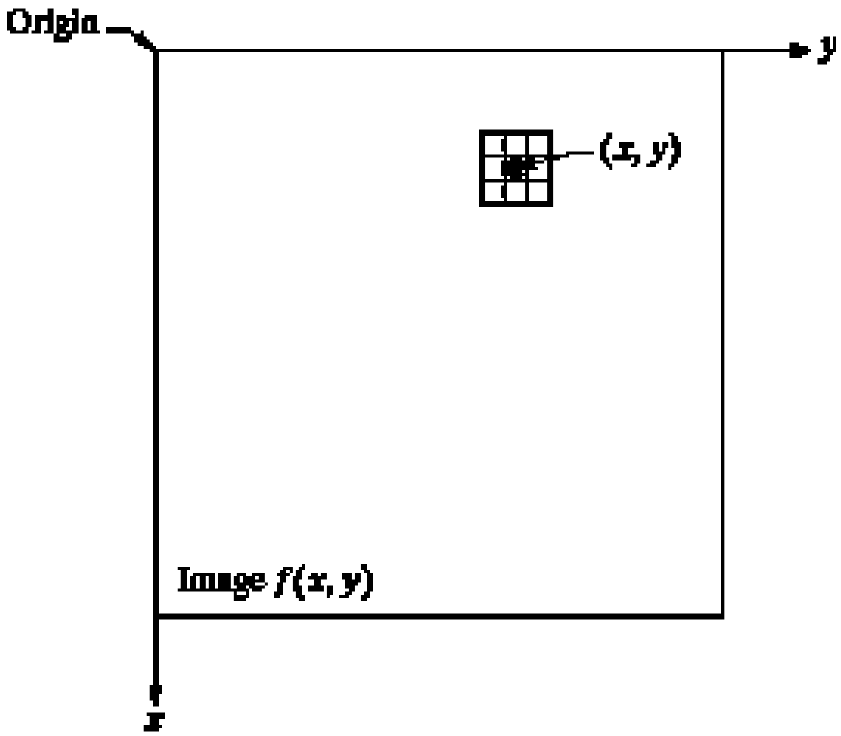
2 domains

- Spatial Domain : (image plane)
  - Techniques are based on direct manipulation of pixels in an image
- Frequency Domain :
  - Techniques are based on modifying the Fourier transform of an image
- There are some enhancement techniques based on various combinations of methods from these two categories.

Good images

- For human visual
  - The visual evaluation of image quality is a highly subjective process.
  - It is hard to standardize the definition of a good image.
- For machine perception
  - The evaluation task is easier.
  - A good image is one which gives the best machine recognition results.
- A certain amount of trial and error usually is required before a particular image enhancement approach is selected.

Spatial Domain



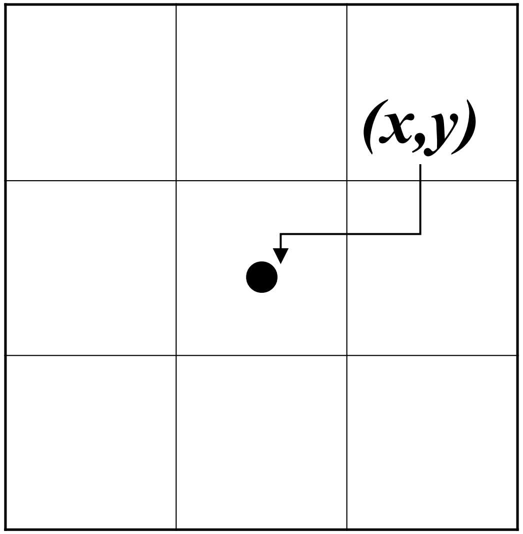
- Procedures that operate directly on pixels.

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

where

- $f(x,y)$ is the input image
- $g(x,y)$ is the processed image
- T is an operator on f defined over some neighborhood of (x,y)

Mask/Filter



- Neighborhood of a point (x,y) can be defined by using a square/rectangular (common used) or circular subimage area centered at (x,y)

- The center of the subimage is moved from pixel to pixel starting at the top of the corner

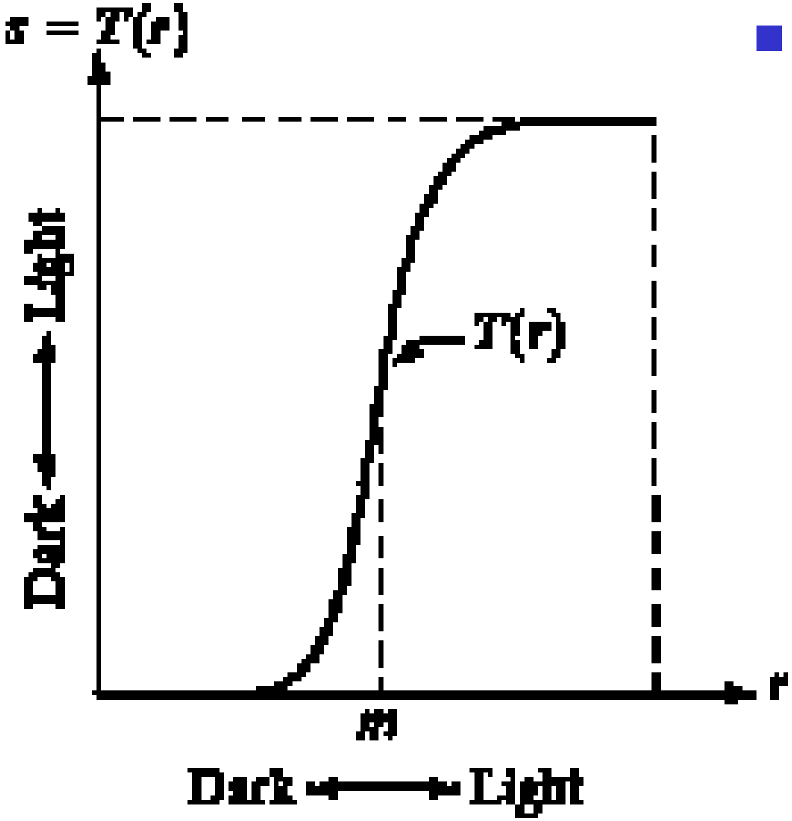
Point Processing

- Neighborhood = 1x1 pixel
- g depends on only the value of f at (x,y)
- T = gray level (or intensity or mapping) transformation function

$$s = T(r)$$

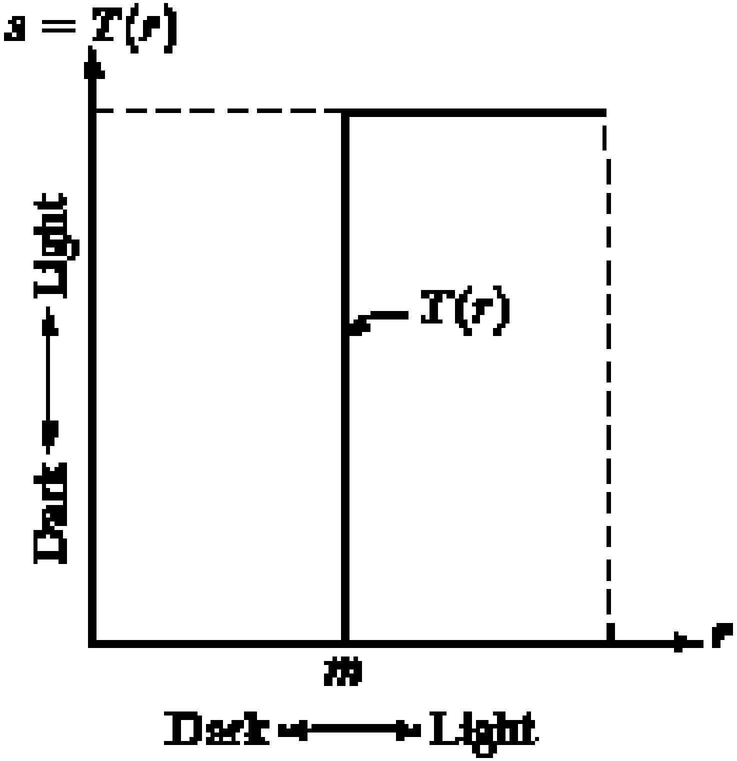
- Where
  - r = gray level of $f(x,y)$
  - s = gray level of $g(x,y)$

Contrast Stretching



- Produce higher contrast than the original by
  - darkening the levels below m in the original image
  - Brightening the levels above m in the original image

Thresholding

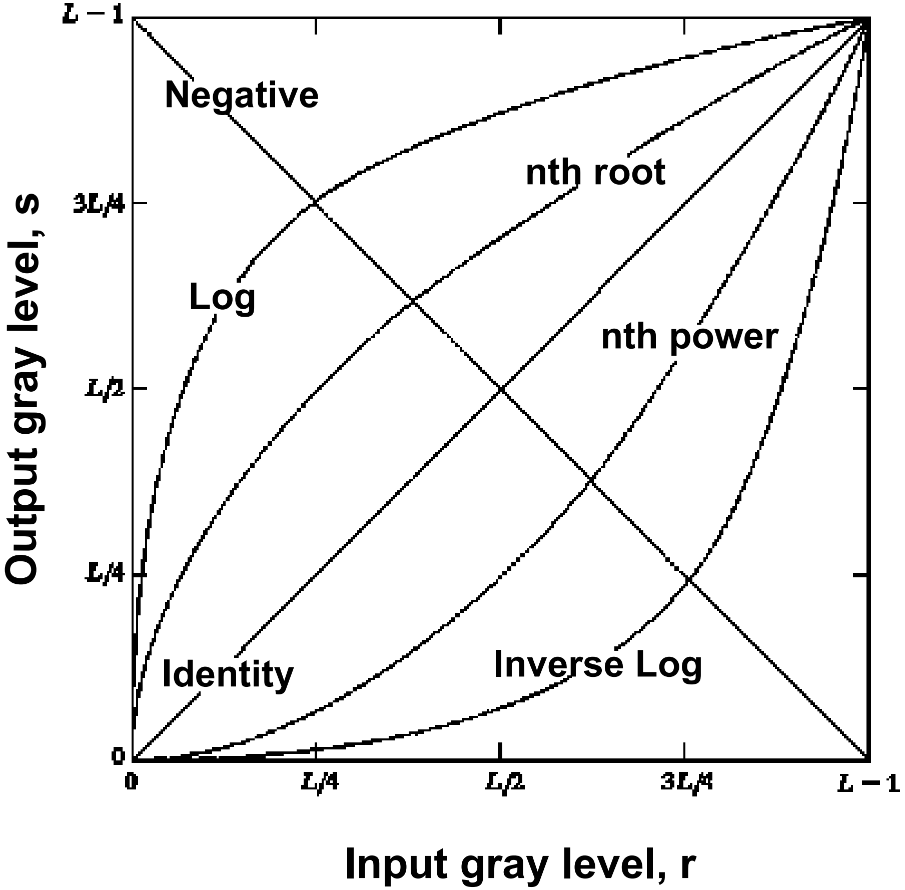


- Produce a two-level (binary) image

Mask Processing or Filter

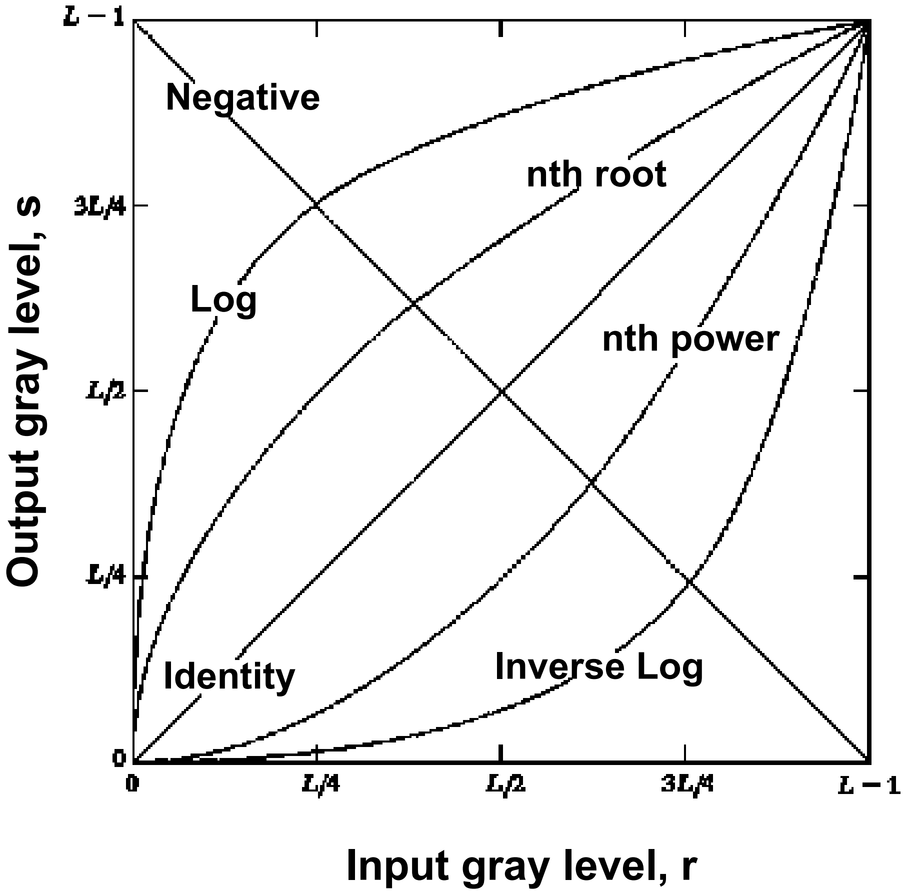
- Neighborhood is bigger than 1x1 pixel
- Use a function of the values of f in a predefined neighborhood of (x,y) to determine the value of g at (x,y)
- The value of the mask coefficients determine the nature of the process
- Used in techniques
  - Image Sharpening
  - Image Smoothing

3 basic gray-level transformation functions



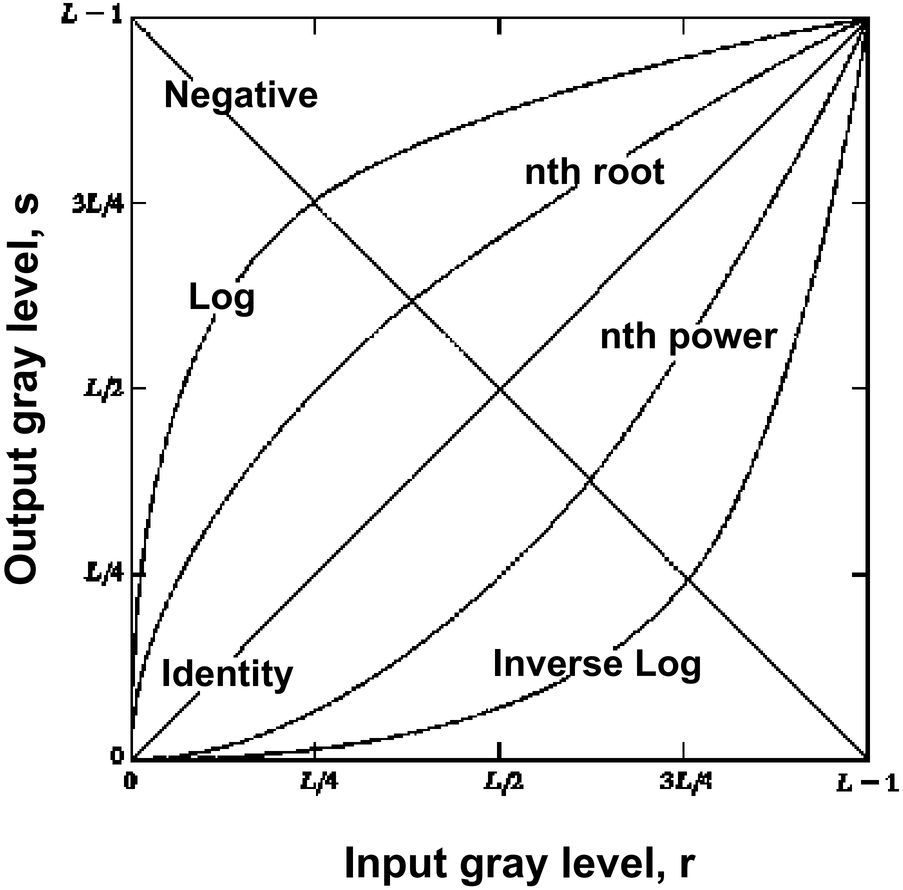
- Linear function
  - Negative and identity transformations
- Logarithm function
  - Log and inverse-log transformation
- Power-law function
  - n^{th} power and n^{th} root transformations

Identity function



- Output intensities are identical to input intensities.
- Is included in the graph only for completeness.

Image Negatives

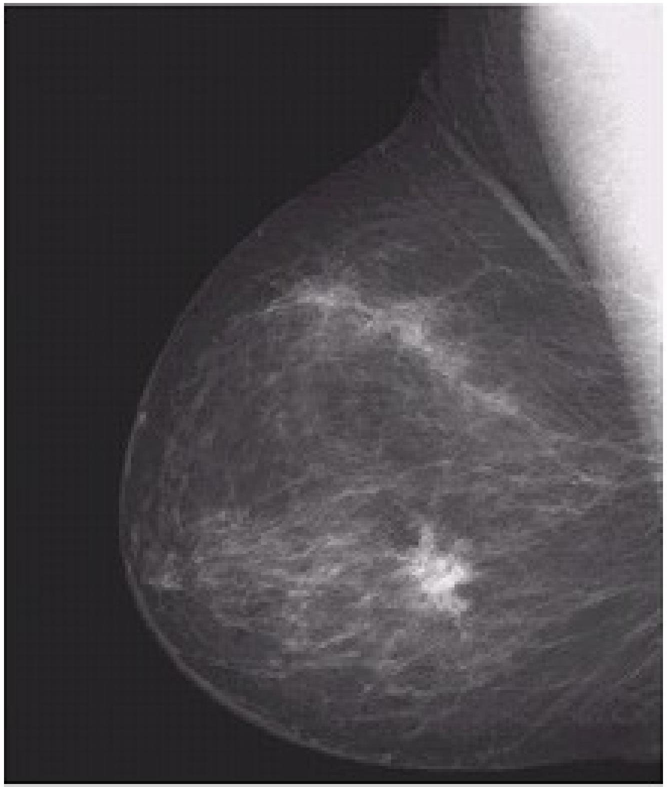


- An image with gray level in the range ***[0, L-1]*** where ***$L = 2^n$*** ; n = 1, 2…
- Negative transformation :

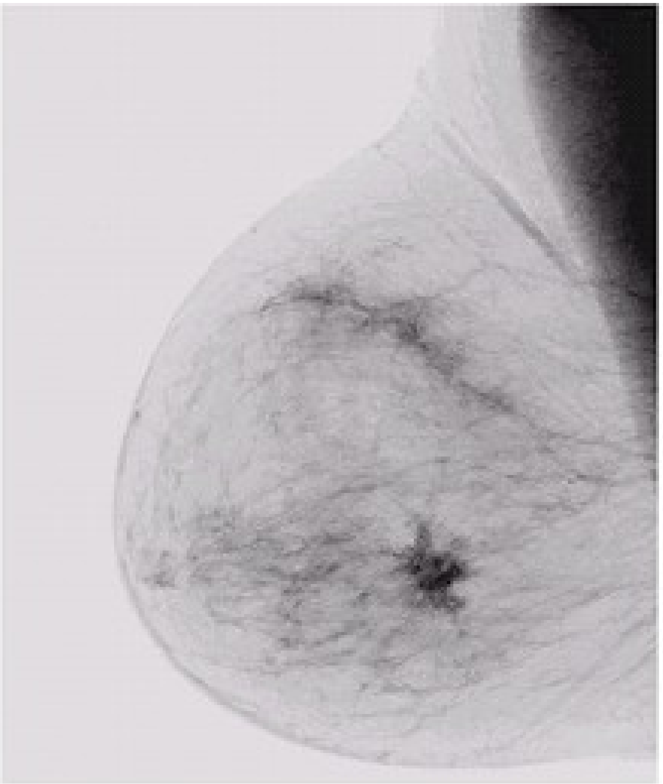
$$s = L - 1 - r$$

- Reversing the intensity levels of an image.
- Suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area dominant in size.

Example of Negative Image

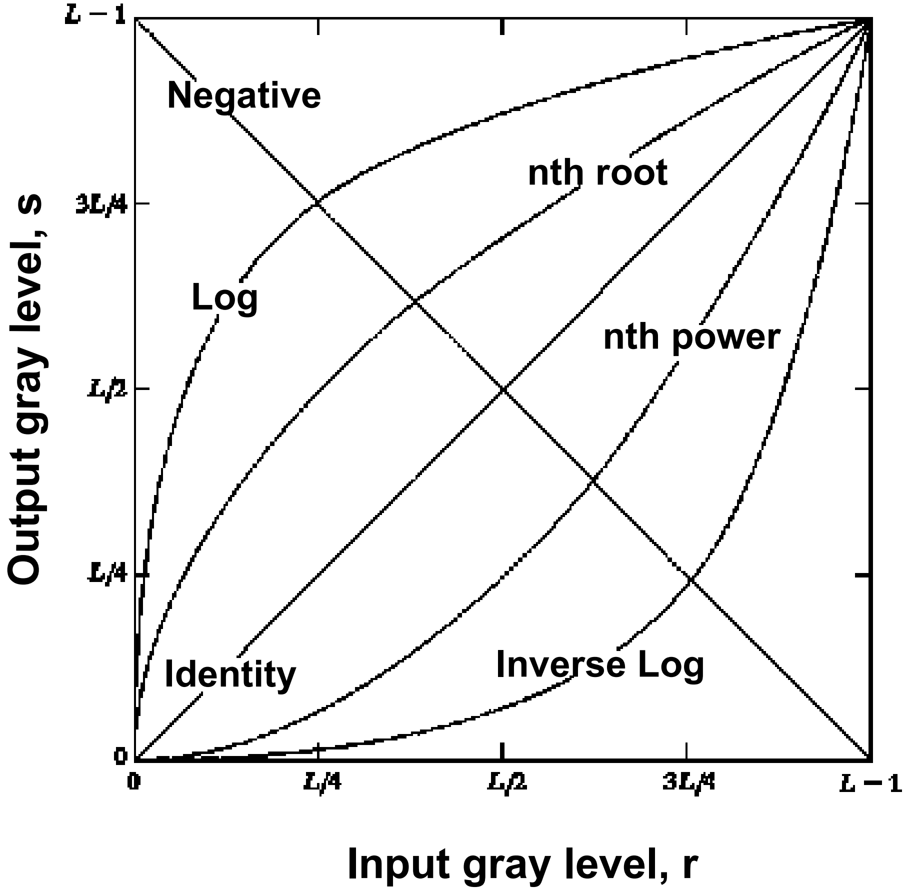


Original mammogram showing a small lesion of a breast



Negative Image : gives a better vision to analyze the image

Log Transformations



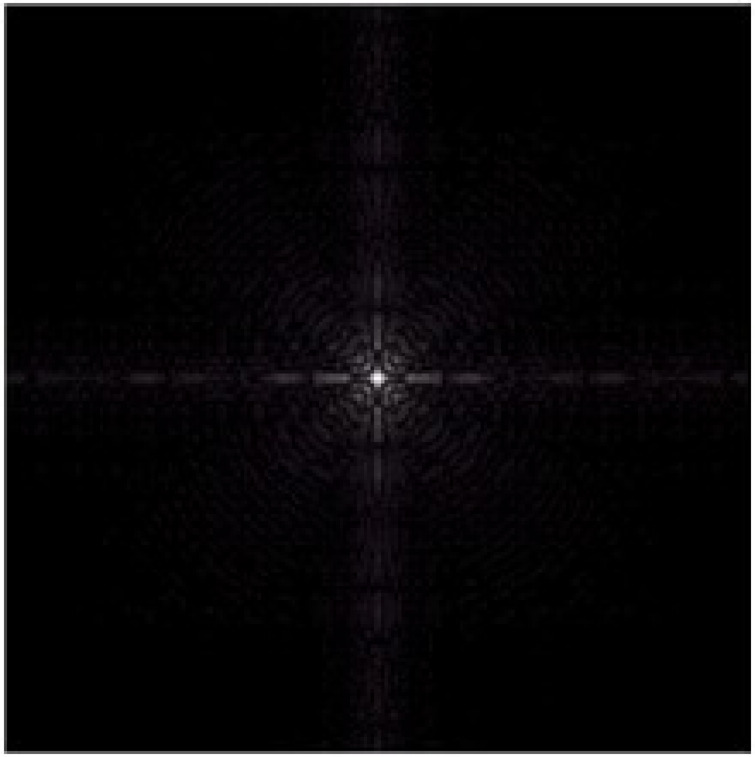
$$s = c \log(1+r)$$

- c is a constant and $r \geq 0$
- Log curve maps a narrow range of low gray-level values in the input image into a wider range of output levels.
- Used to expand the values of dark pixels in an image while compressing the higher-level values.

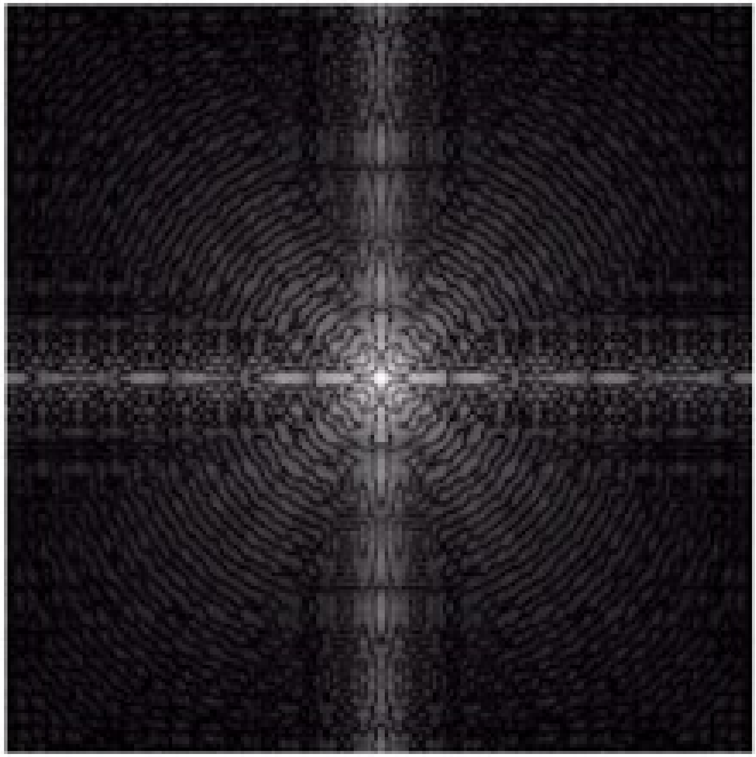
Log Transformations

- It compresses the dynamic range of images with large variations in pixel values
- Example of image with dynamic range: Fourier spectrum image
- It can have intensity range from 0 to 10^6 or higher.
- We can't see the significant degree of detail as it will be lost in the display.

Example of Logarithm Image



Fourier Spectrum with range = 0 to 1.5 x 10^6

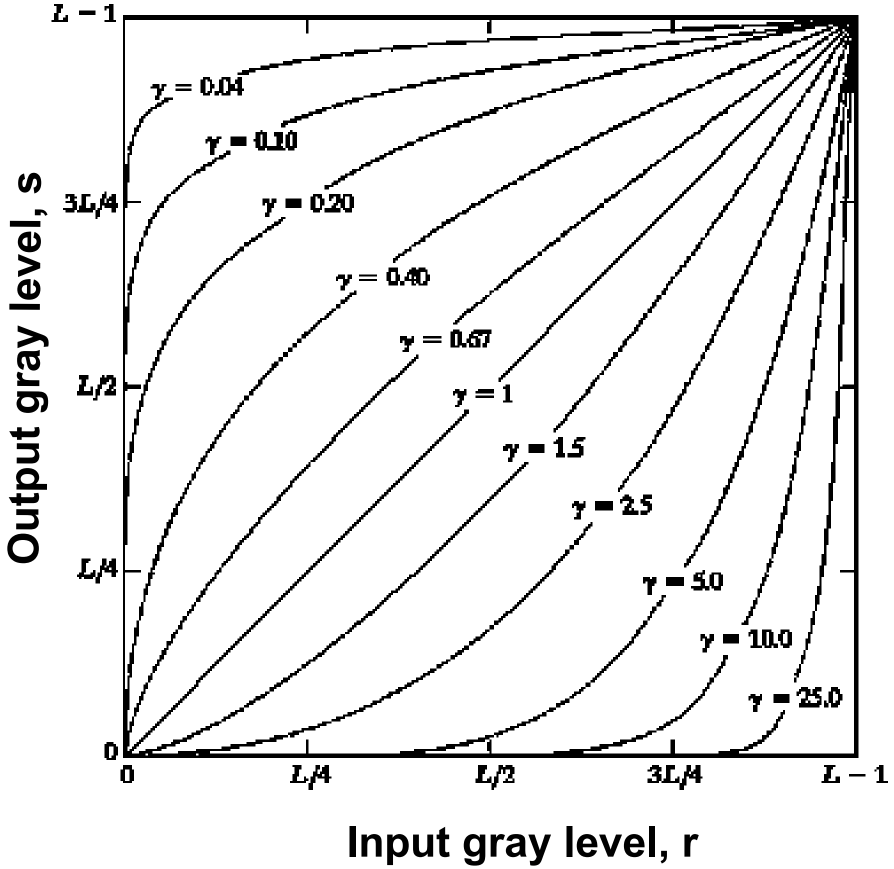


Result after apply the log transformation with c = 1, range = 0 to 6.2

Inverse Logarithm Transformations

- Do opposite to the Log Transformations
- Used to expand the values of high pixels in an image while compressing the darker-level values.

Power-Law Transformations

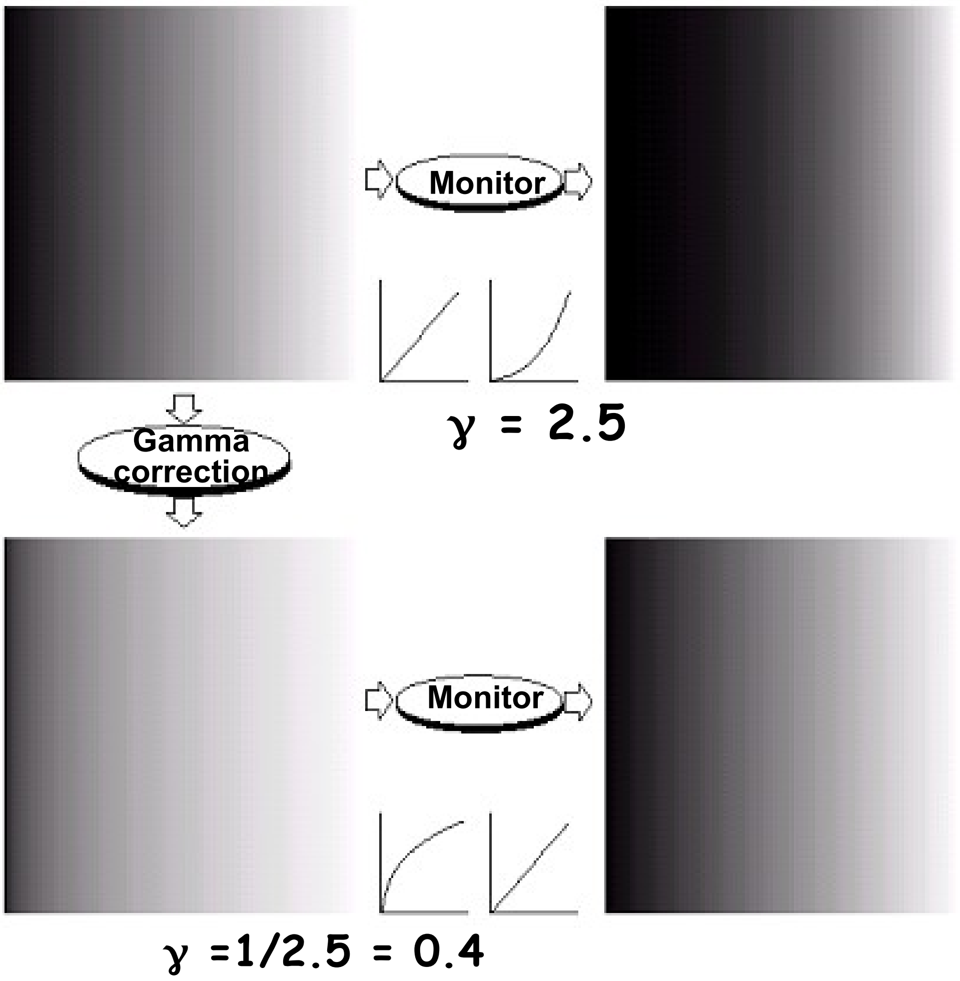


Plots of $s = cr^{\gamma}$ for various values of γ
(c = 1 in all cases)

$$s = cr^{\gamma}$$

- c and γ are positive constants
- Power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.
- c = γ = 1 → Identity function

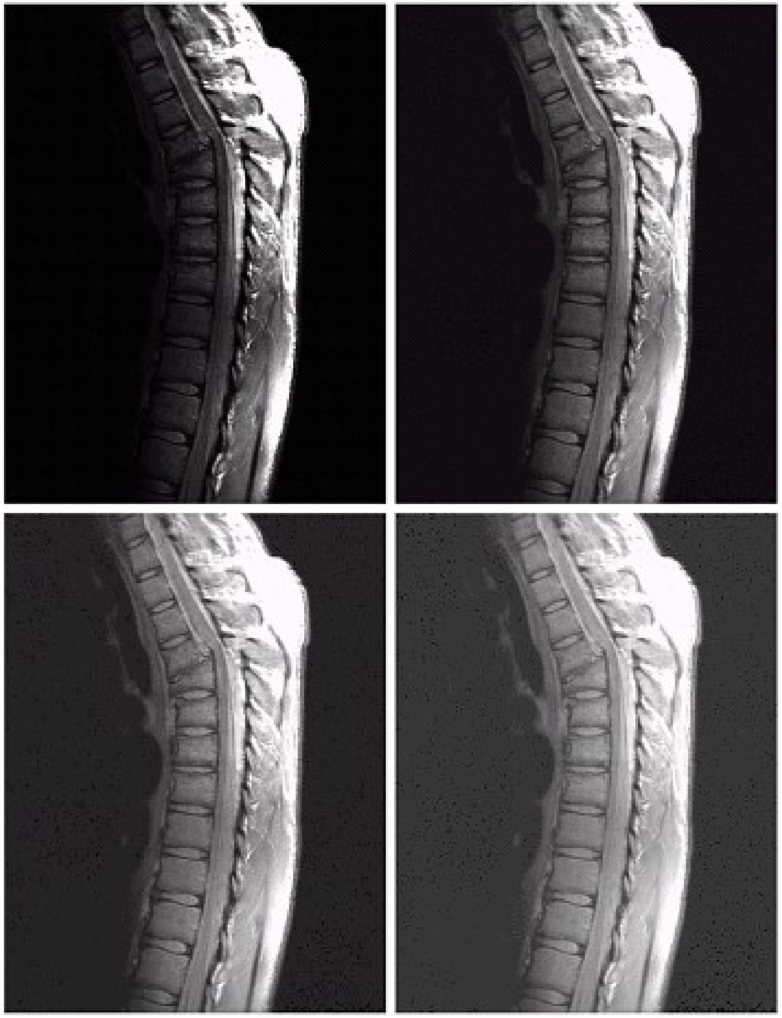
Gamma correction



- Cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with γ varying from 1.8 to 2.5
- The picture will become darker.
- Gamma correction is done by preprocessing the image before inputting it to the monitor with $s = cr^{1/\gamma}$

Another example : MRI

a	b
c	d



(a) a magnetic resonance image of an upper thoracic human spine with a fracture dislocation and spinal cord impingement

- The picture is predominately dark
- An expansion of gray levels are desirable needs $\gamma < 1$

(b) result after power-law transformation with $\gamma = 0.6$, c=1

(c) transformation with $\gamma = 0.4$ (best result)

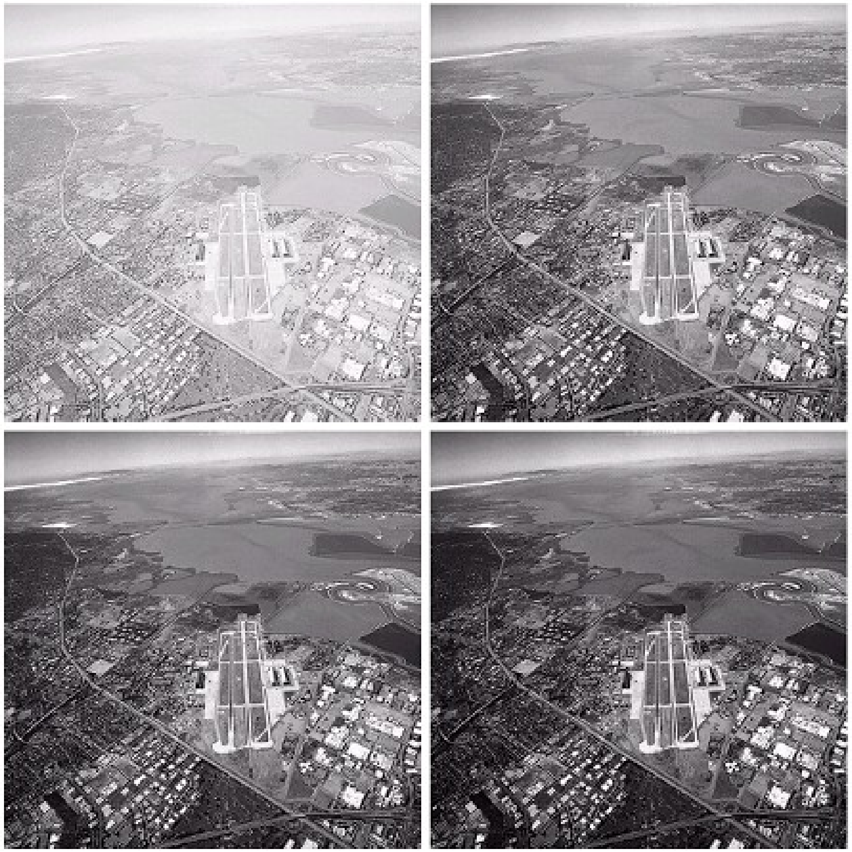
(d) transformation with $\gamma = 0.3$ (under acceptable level)

Effect of decreasing gamma

- When the γ is reduced too much, the image begins to reduce contrast to the point where the image started to have very slight “wash-out” look, especially in the background

Another example

a	b
c	d



(a) image has a washed-out appearance, it needs a compression of gray levels needs $\gamma > 1$

(b) result after power-law transformation with γ = 3.0 (suitable)

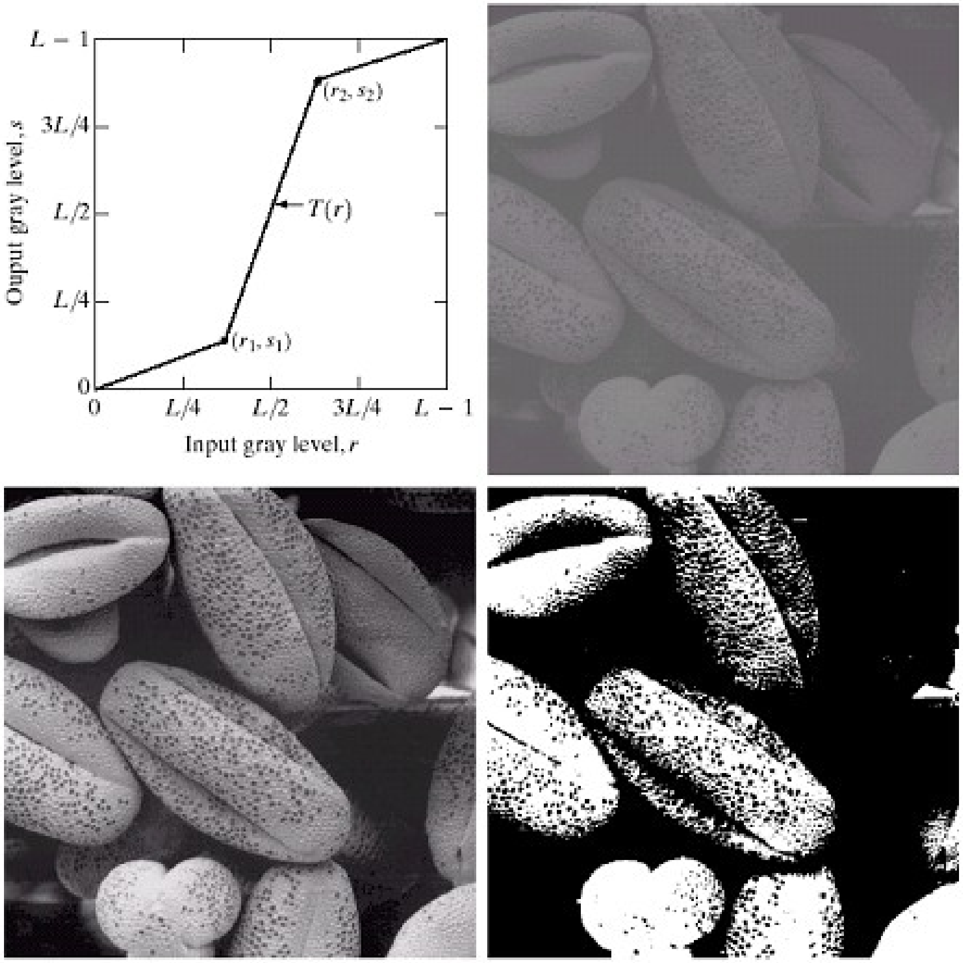
(c) transformation with γ = 4.0 (suitable)

(d) transformation with γ = 5.0 (high contrast, the image has areas that are too dark, some detail is lost)

Piecewise-Linear Transformation Functions

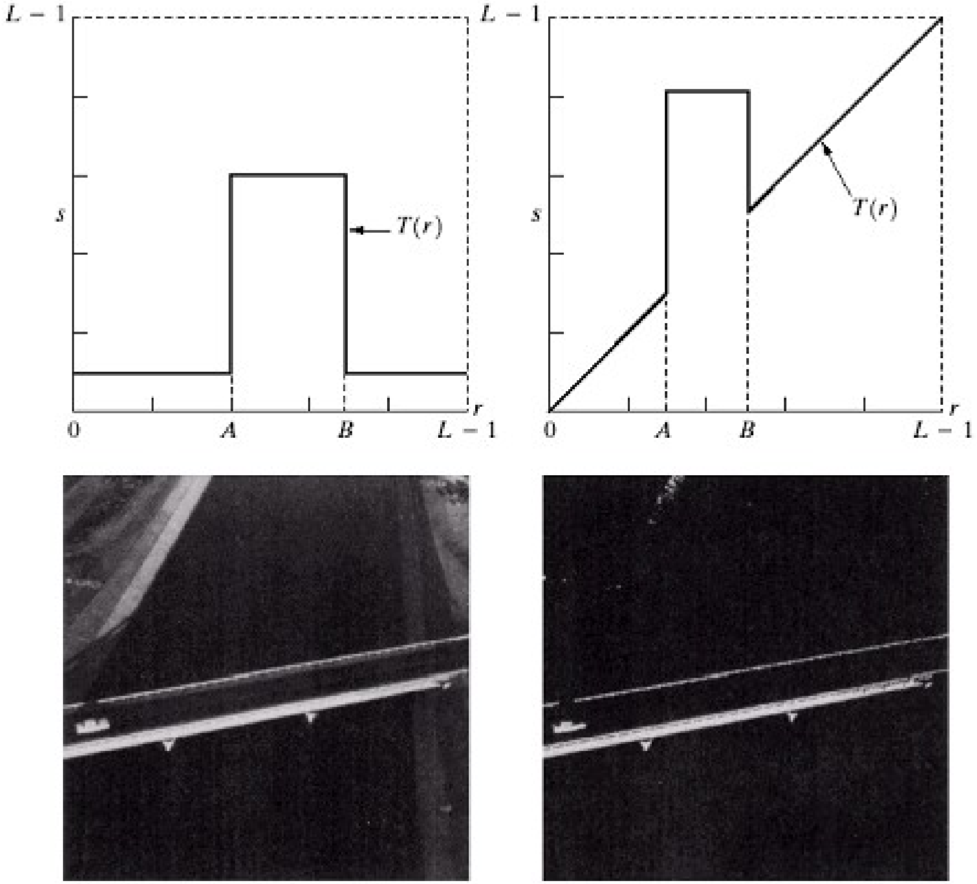
- Advantage:
  - The form of piecewise functions can be arbitrarily complex
- Disadvantage:
  - Their specification requires considerably more user input

Contrast Stretching



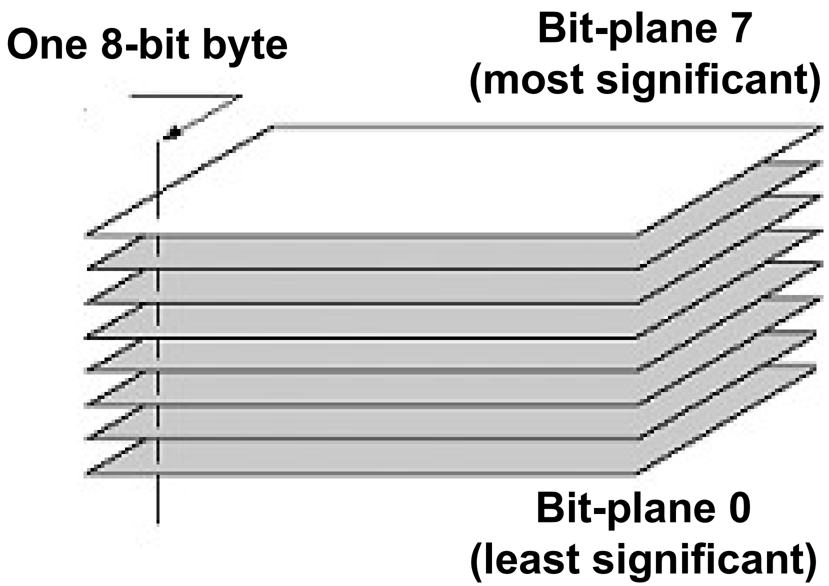
- increase the dynamic range of the gray levels in the image
- (b) a low-contrast image : result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture of image acquisition
- (c) result of contrast stretching: $(r_1,s_1) = (r_{min},0)$ and $(r_2,s_2) = (r_{max},L-1)$
- (d) result of thresholding

Gray-level slicing



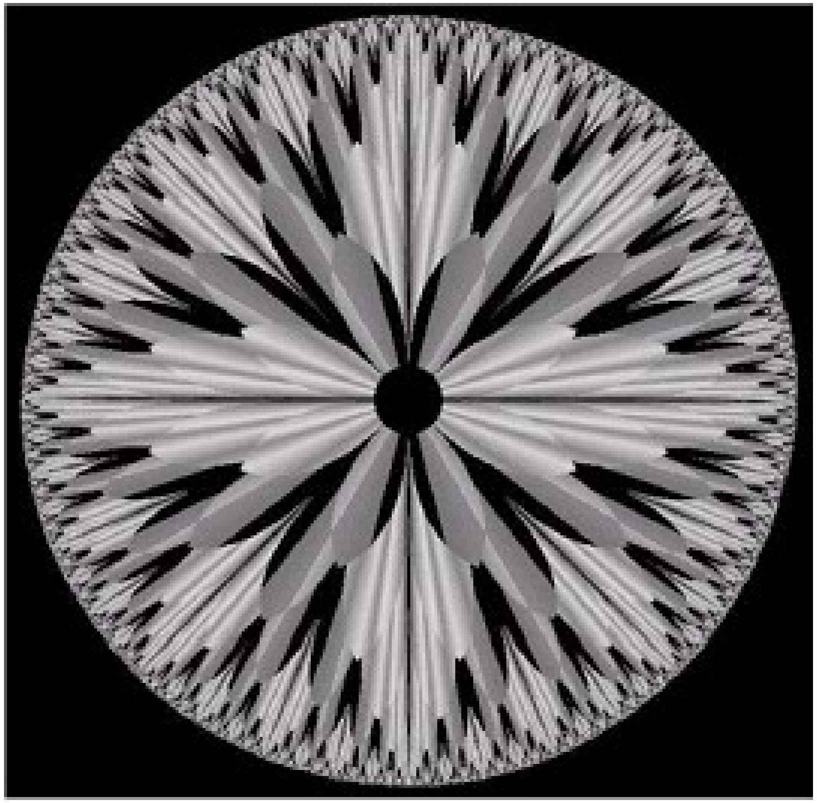
- Highlighting a specific range of gray levels in an image
  - Display a high value of all gray levels in the range of interest and a low value for all other gray levels
- (a) transformation highlights range [A,B] of gray level and reduces all others to a constant level
- (b) transformation highlights range [A,B] but preserves all other levels

Bit-plane slicing



- Highlighting the contribution made to total image appearance by specific bits
- Suppose each pixel is represented by 8 bits
- Higher-order bits contain the majority of the visually significant data
- Useful for analyzing the relative importance played by each bit of the image

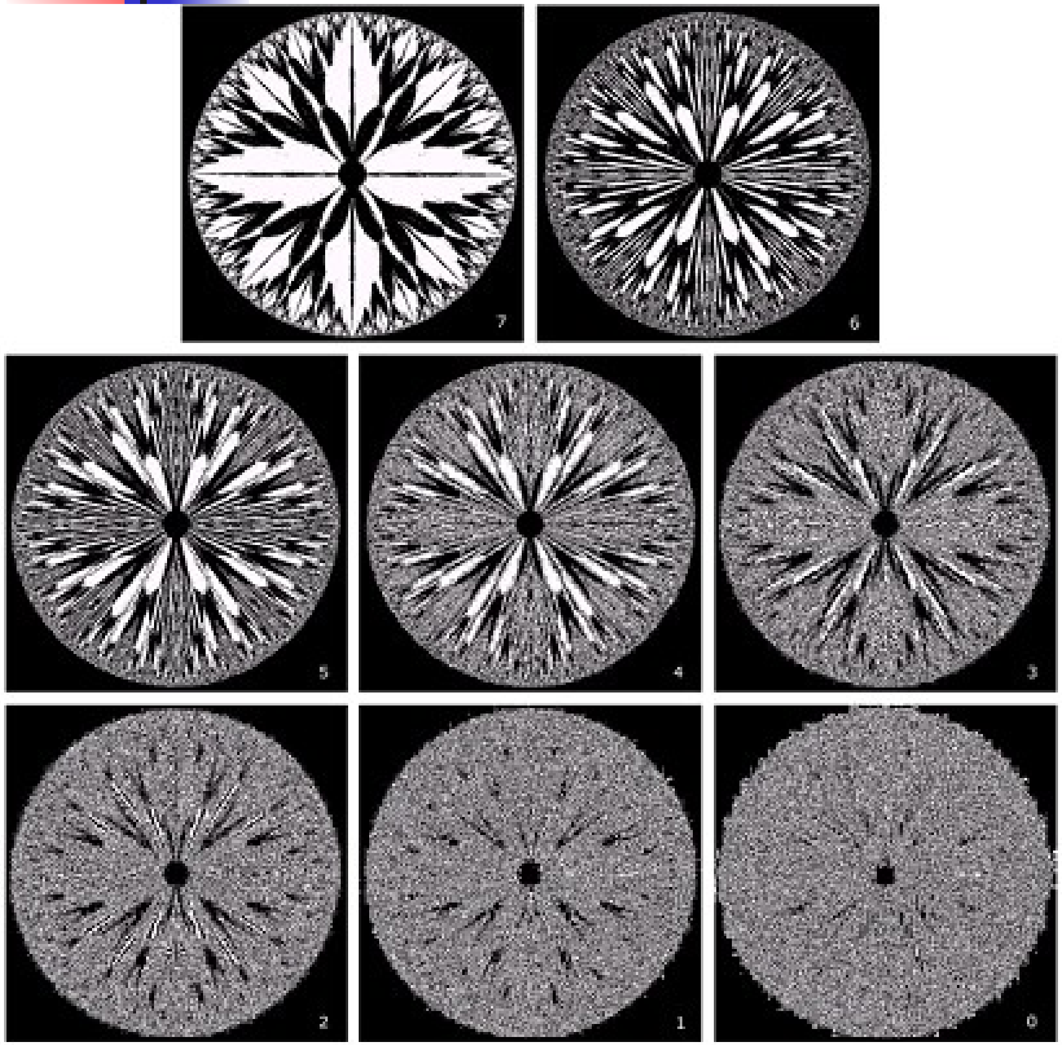
Example



An 8-bit fractal image

- The (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation.
  - Map all levels between 0 and 127 to 0
  - Map all levels between 129 and 255 to 255

8 bit planes



<table>
<tr><td colspan="3">Bit-plane 7</td><td colspan="3">Bit-plane 6</td></tr>
<tr><td colspan="2">Bit-plane 5</td><td colspan="2">Bit-plane 4</td><td colspan="2">Bit-plane 3</td></tr>
<tr><td colspan="2">Bit-plane 2</td><td colspan="2">Bit-plane 1</td><td colspan="2">Bit-plane 0</td></tr>
</table>

Histogram Processing

- 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 : the kth gray level
  - n_k : the number of pixels in the image having gray level r_k
  - $h(r_k)$: histogram of a digital image with gray levels r_k

Normalized Histogram

- dividing each of histogram at gray level r_k by the total number of pixels in the image, n

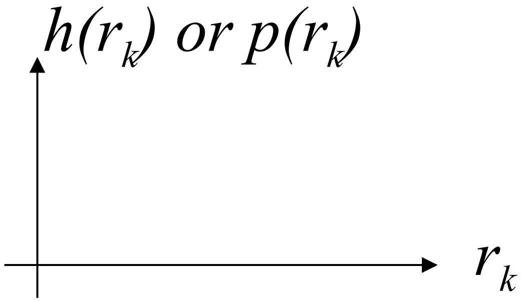
$$p(r_k) = n_k / n$$

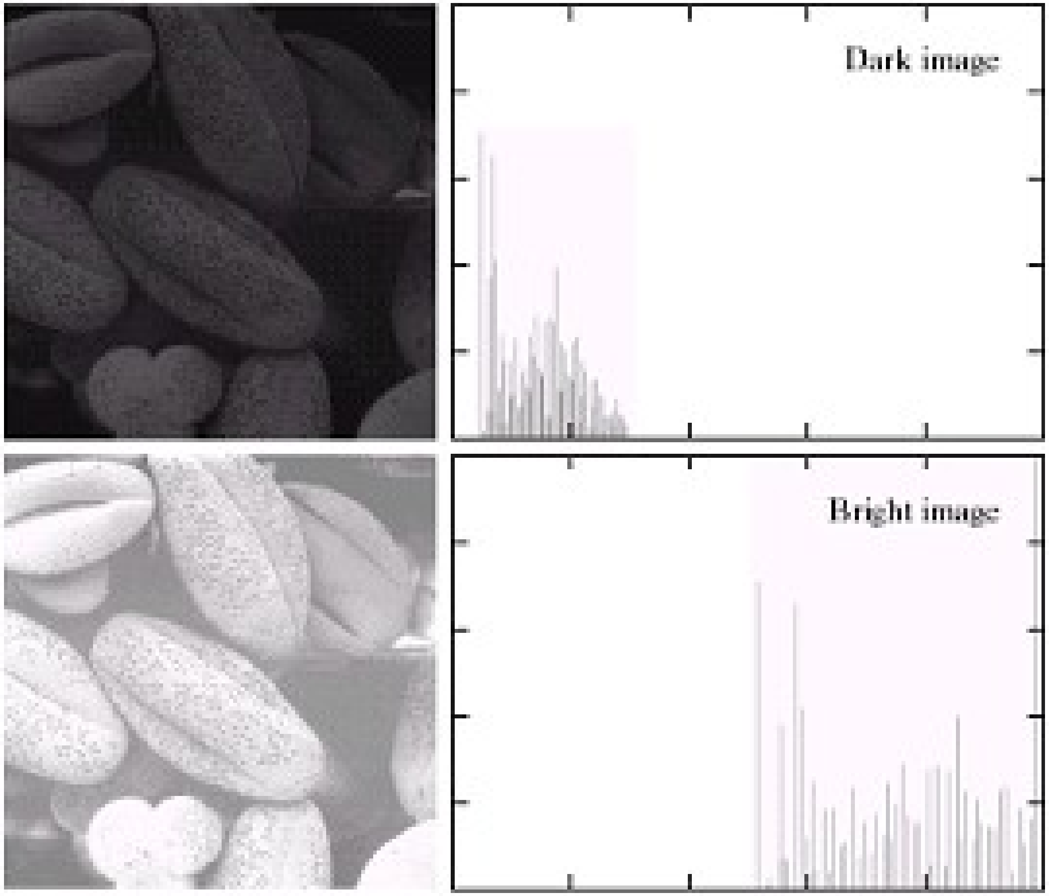
- For k = 0,1,…,L-1
- $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k
- The sum of all components of a normalized histogram is equal to 1

Histogram Processing

- Basic for numerous spatial domain processing techniques
- Used effectively for image enhancement
- Information inherent in histograms also is useful in image compression and segmentation

Example





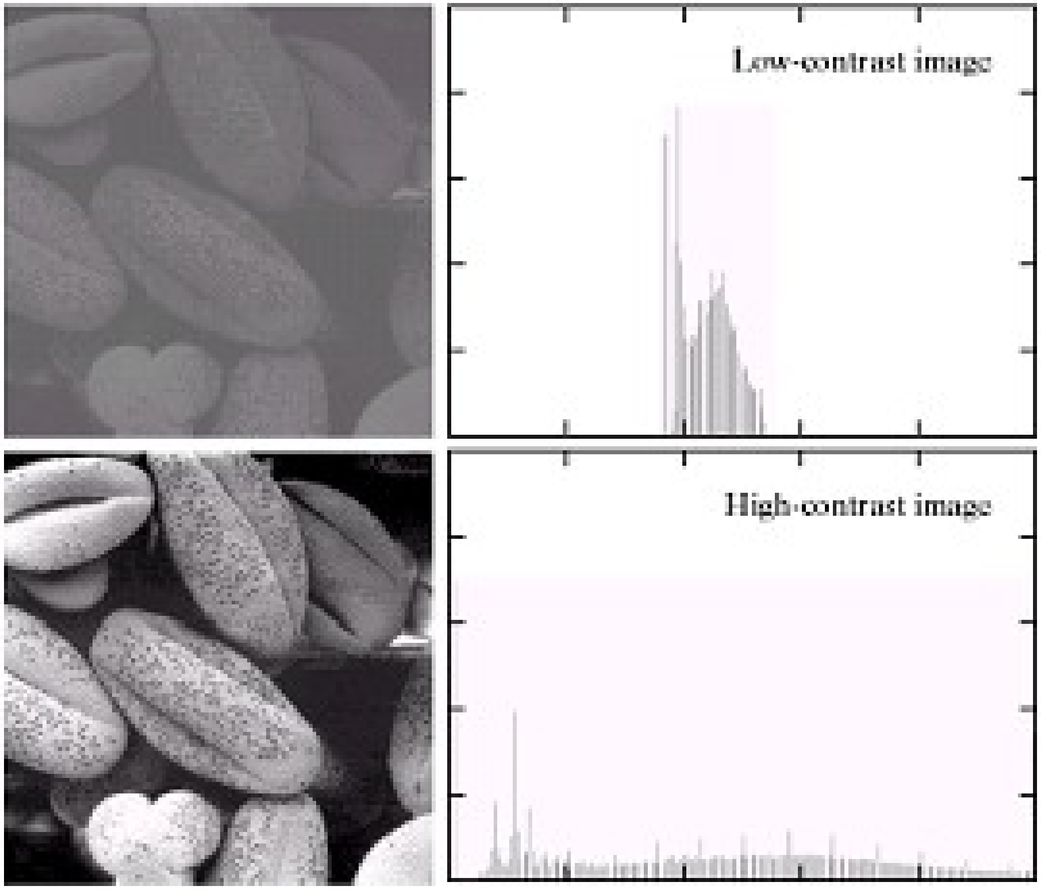
Dark image

Components of histogram are concentrated on the low side of the gray scale.

Bright image

Components of histogram are concentrated on the high side of the gray scale.

Example



Low-contrast image

histogram is narrow and centered toward the middle of the gray scale

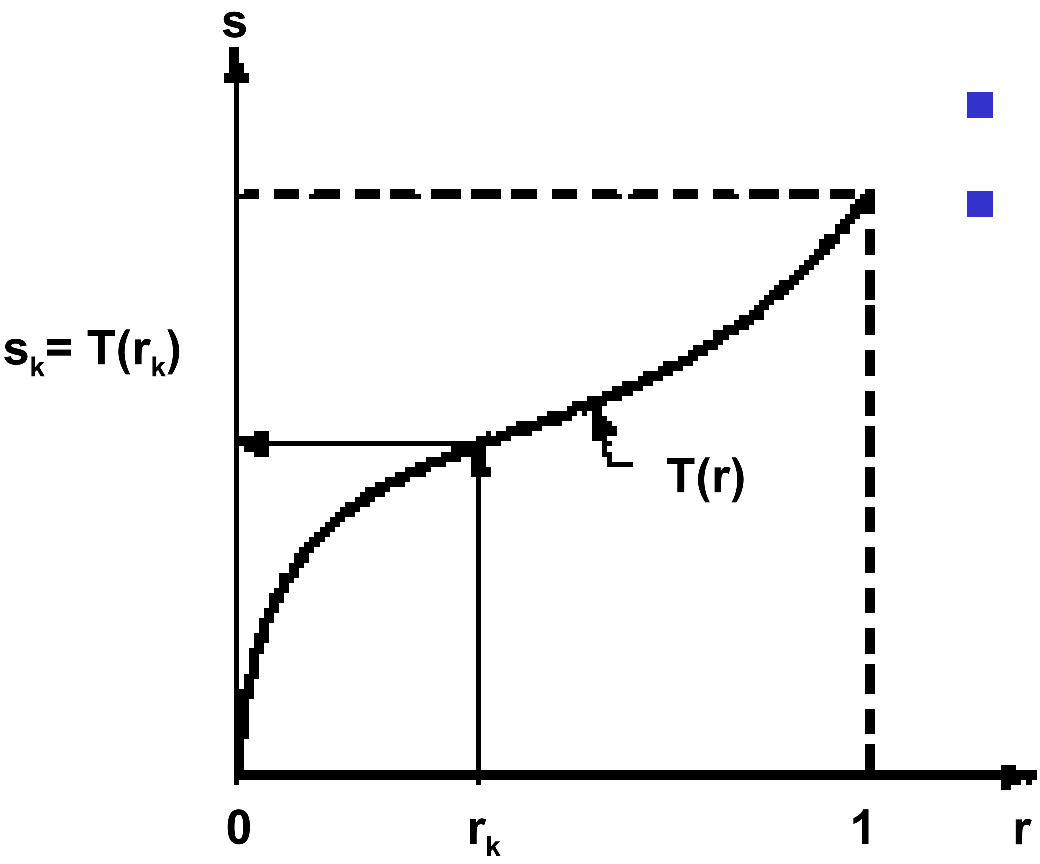
High-contrast image

histogram covers broad range of the gray scale and the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others

Histogram Equalization

- As the low-contrast image's histogram is narrow and centered toward the middle of the gray scale, if we distribute the histogram to a wider range the quality of the image will be improved.
- We can do it by adjusting the probability density function of the original histogram of the image so that the probability spread equally

Histogram transformation



$$s = T(r)$$

- Where $0 \le r \le 1$
- T(r) satisfies
  - (a). T(r) is single-valued and monotonically increasingly in the interval $0 \le r \le 1$
  - (b). $0 \le T(r) \le 1$ for $0 \le r \le 1$

2 Conditions of T(r)

- Single-valued (one-to-one relationship) guarantees that the inverse transformation will exist
- Monotonicity condition preserves the increasing order from black to white in the output image thus it won't cause a negative image
- $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$ guarantees that the output gray levels will be in the same range as the input levels.
- The inverse transformation from s back to r is

$$r = T^{-1}(s) \quad ; 0 \leq s \leq 1$$

Probability Density Function

- The gray levels in an image may be viewed as random variables in the interval [0,1]
- PDF is one of the fundamental descriptors of a random variable

Random Variables

- Random variables often are a source of confusion when first encountered.
- This need not be so, as the concept of a random variable is in principle quite simple.

Random Variables

- A random variable, x, is a real-valued function defined on the events of the sample space, S.
- In words, for each event in S, there is a real number that is the corresponding value of the random variable.
- Viewed yet another way, **a random variable maps each event in S onto the real line**.
- That is it. A simple, straightforward definition.

Random Variables

- Part of the confusion often found in connection with random variables is the fact that **they are functions**.
- The notation also is partly responsible for the problem.

Random Variables

- In other words, although typically the notation used to denote a random variable is as we have shown it here, x, or some other appropriate variable,
- to be strictly formal, a random variable should be written as a function $x(\cdot)$ where the argument is a specific event being considered.

Random Variables

- However, this is seldom done, and, in our experience, trying to be formal by using function notation complicates the issue more than the clarity it introduces.
- Thus, we will opt for the less formal notation, with the warning that it must be keep clearly in mind that random variables are functions.

Random Variables

- Example:
  - Consider the experiment of drawing a single card from a standard deck of 52 cards.
  - Suppose that we define the following events. A: a heart; B: a spade; C: a club; and D: a diamond, so that S = {A, B, C, D}.
  - A random variable is easily defined by letting x = 1 represent event A, x = 2 represent event B, and so on.

Random Variables

- As a second illustration,
  - consider the experiment of throwing a single die and observing the value of the up-face.
  - We can define a random variable as the numerical outcome of the experiment (i.e., 1 through 6), but there are many other possibilities.
  - For example, a binary random variable could be defined simply by letting x = 0 represent the event that the outcome of throw is an even number and x = 1 otherwise.

Random Variables

- Note
  - the important fact in the examples just given that the probability of the events have not changed;
  - all a random variable does is map events onto the real line.

Random Variables

- Thus far we have been concerned with random variables whose values are discrete.
- To handle continuous random variables we need some additional tools.
- In the discrete case, the probabilities of events are numbers between 0 and 1.

Random Variables

- When dealing with continuous quantities (which are not denumerable) we can no longer talk about the "probability of an event" because that probability is zero.
- This is not as unfamiliar as it may seem.

Random Variables

- For example,
  - given a continuous function we know that the area of the function between two limits a and b is the integral from a to b of the function.
  - However, the area at a point is zero because the integral from,say, a to a is zero.
  - We are dealing with the same concept in the case of continuous random variables.

Random Variables

- Thus, instead of talking about the probability of a specific value, we talk about the probability that the value of the random variable lies in a specified range.
- In particular, we are interested in the probability that the random variable is less than or equal to (or, similarly, greater than or equal to) a specified constant a.
- We write this as

$$F(a) = P(x \leq a)$$

Random Variables

- If this function is given for all values of a (i.e., $-\infty < a < \infty$), then the values of random variable x have been defined.
- Function F is called the cumulative probability distribution function or simply the cumulative distribution function (cdf).
- The shortened term distribution function also is used.

Random Variables

- Observe that the notation we have used makes no distinction between a random variable and the values it assumes.
- If confusion is likely to arise, we can use more formal notation in which we let capital letters denote the random variable and lowercase letters denote its values.
- For example, the cdf using this notation is written as

$$F_X(x) = P(X \leq x)$$

Random Variables

- When confusion is not likely, the cdf often is written simply as $F(x)$.
- This notation will be used in the following discussion when speaking generally about the cdf of a random variable.

Random Variables

- Due to the fact that it is a probability, the cdf has the following properties:

1. $F(-\infty) = 0$
2. $F(\infty) = 1$
3. $0 \leq F(x) \leq 1$
4. $F(x_1) \leq F(x_2)$ *if* $x_1 < x_2$
5. $P(x_1 < x \leq x_2) = F(x_2) - F(x_1)$
6. $F(x+) = F(x),$

where $x+ = x + \varepsilon$, with ε being a positive, infinitesimally small number.

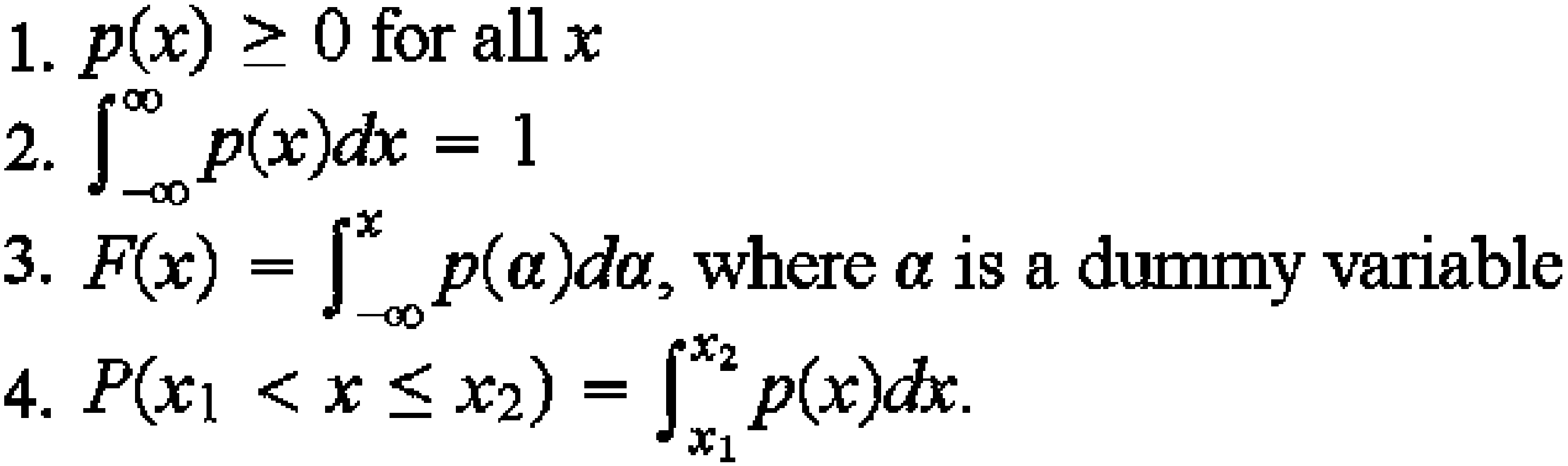
Random Variables

The probability density function (pdf or shortly called density function) of random variable x is defined as the derivative of the cdf:

$$p(x) = \frac{dF(x)}{dx}$$

Random Variables

The pdf satisfies the following properties:



1. $p(x) \geq 0$ for all x

2. $\int_{-\infty}^{\infty} p(x)dx = 1$

3. $F(x) = \int_{-\infty}^{x} p(\alpha)d\alpha$, where α is a dummy variable

4. $P(x_1 < x \leq x_2) = \int_{x_1}^{x_2} p(x)dx.$

Random Variables

- The preceding concepts are applicable to discrete random variables.
- In this case, there is a finite no. of events and we talk about probabilities, rather than probability density functions.
- Integrals are replaced by summations and, sometimes, the random variables are subscripted.
- For example, in the case of a discrete variable with N possible values we would denote the probabilities by $P(x_i)$, $i=1, 2,\ldots, N$.

Random Variables

- If a random variable x is transformed by a monotonic transformation function $T(x)$ to produce a new random variable y,
- the probability density function of y can be obtained from knowledge of $T(x)$ and the probability density function of x, as follows:

$$p_y(y) = p_x(x)\left|\frac{dx}{dy}\right|$$

where the vertical bars signify the absolute value.

Random Variables

- A function $T(x)$ is monotonically increasing if $T(x_1) < T(x_2)$ for $x_1 < x_2$, and
- A function $T(x)$ is monotonically decreasing if $T(x_1) > T(x_2)$ for $x_1 < x_2$.
- The preceding equation is valid if $T(x)$ is an increasing or decreasing monotonic function.

Applied to Image

- Let
  - $p_r(r)$ denote the PDF of random variable r
  - $p_s(s)$ denote the PDF of random variable s
- If $p_r(r)$ and $T(r)$ are known and $T^{-1}(s)$ satisfies condition (a) then $p_s(s)$ can be obtained using a formula :

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

Applied to Image

The PDF of the transformed variable s

is determined by

the gray-level PDF of the input image

and by

the chosen transformation function

Transformation function

- A transformation function is a cumulative distribution function (CDF) of random variable r :

$$s = T(r) = \int_{0}^{r} p_r(w)\,dw$$

where w is a dummy variable of integration

Note: $T(r)$ depends on $p_r(r)$

Cumulative Distribution function

- CDF is an integral of a probability function (always positive) is the area under the function
- Thus, CDF is always single valued and monotonically increasing
- Thus, CDF satisfies the condition (a)
- We can use CDF as a transformation function

Finding $p_s(s)$ from given $T(r)$

$$\frac{ds}{dr}=\frac{dT(r)}{dr}$$

$$=\frac{d}{dr}\left[\int_0^r p_r(w)dw\right]$$

$$=p_r(r)$$

$$p_s(s)=p_r(r)\left|\frac{dr}{ds}\right|$$

$$=p_r(r)\left|\frac{1}{p_r(r)}\right|$$

$$=1 \quad \text{where } 0\leq s\leq 1$$

Substitute and yield

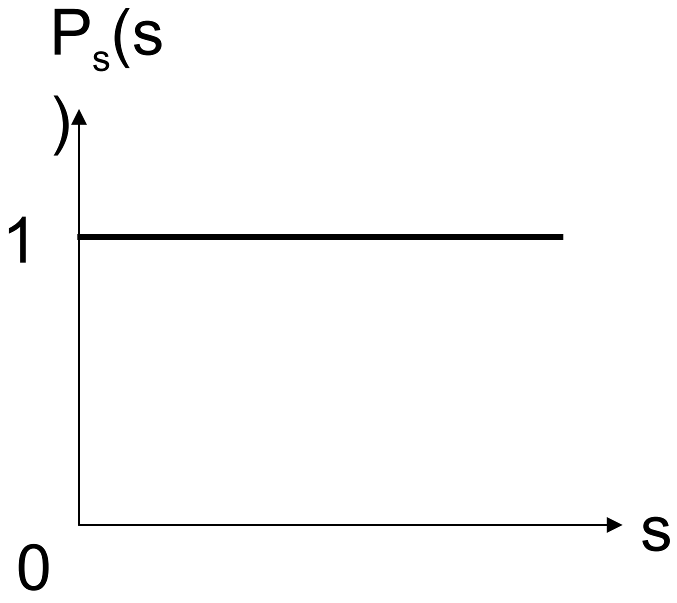
$p_s(s)$

- As $p_s(s)$ is a probability function, it must be zero outside the interval [0,1] in this case because its integral over all values of s must equal 1.
- Called $p_s(s)$ as **a uniform probability density function**
- $p_s(s)$ is always a uniform, independent of the form of $p_r(r)$

$$s = T(r) = \int_0^r p_r(w)\,dw$$

yields

a random variable s characterized by a uniform probability function



Discrete transformation function

- The probability of occurrence of gray level in an image is approximated by

$$p_r(r_k) = \frac{n_k}{n} \qquad where\ k = 0, 1, ..., L\text{-}1$$

- The discrete version of transformation

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{n} \qquad where\ k = 0, 1, ..., L\text{-}1$$

Histogram Equalization

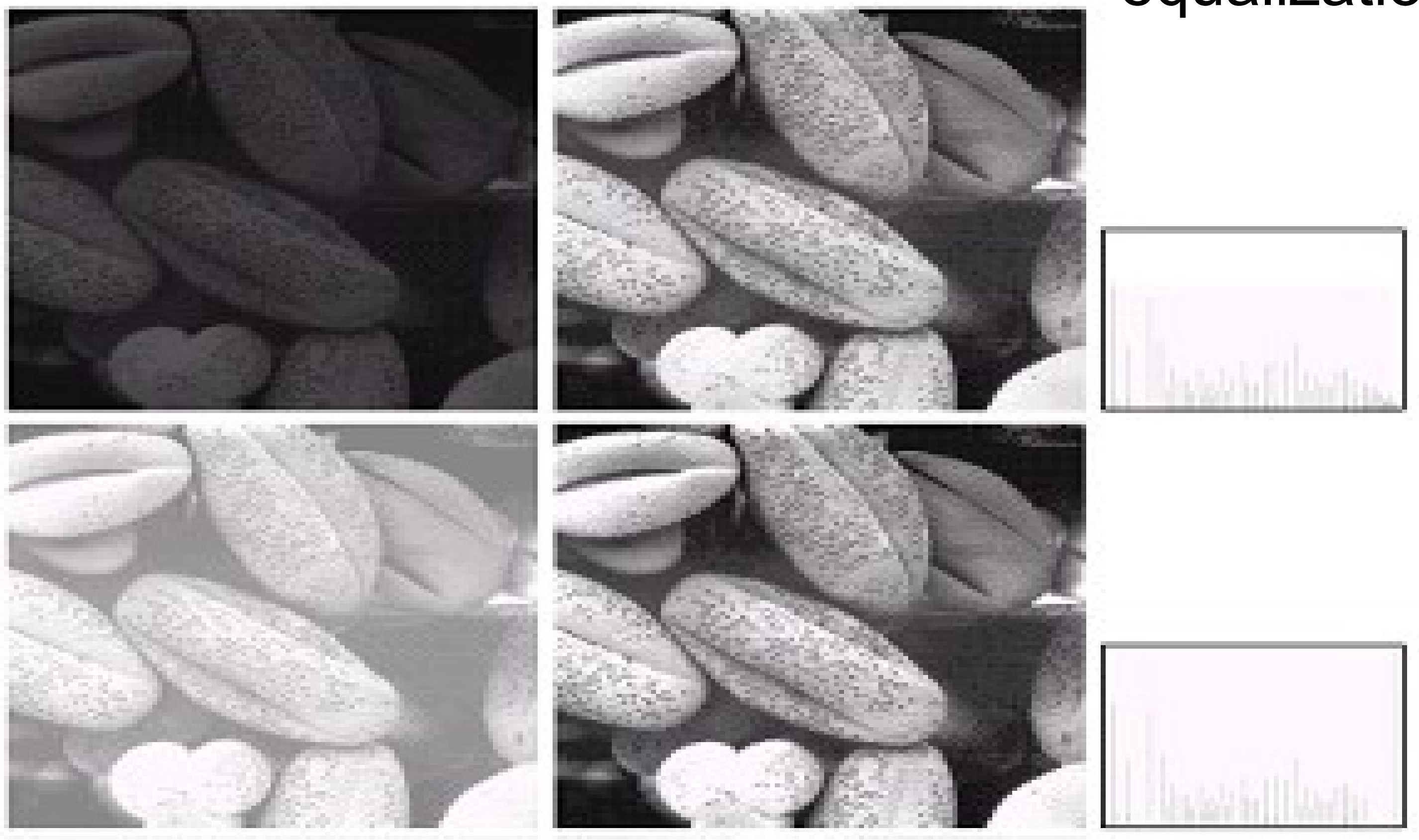
- Thus, an output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k in the output image
- In discrete space, it cannot be proved in general that this discrete transformation will produce the discrete equivalent of a uniform probability density function, which would be a uniform histogram

Example

before

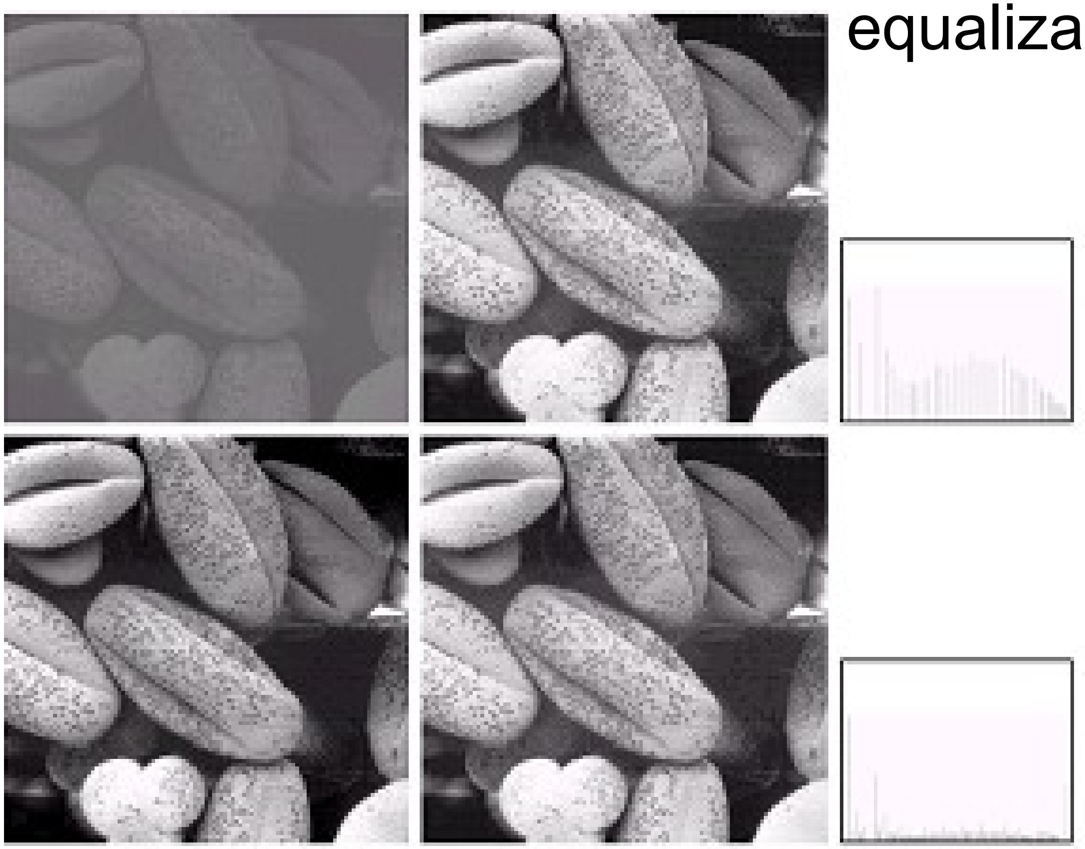
after

Histogram equalization



Example

before after Histogram equalization



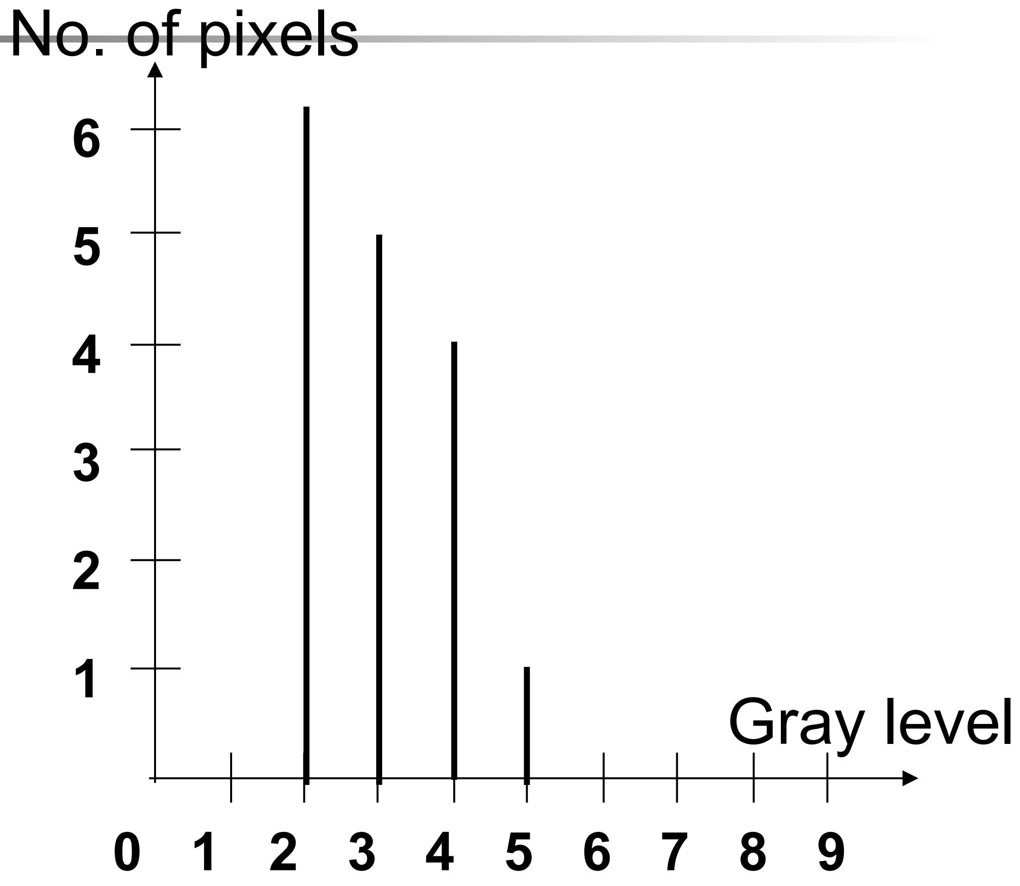
The quality is not improved much because the original image already has a broaden gray-level scale

Example

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



histogram

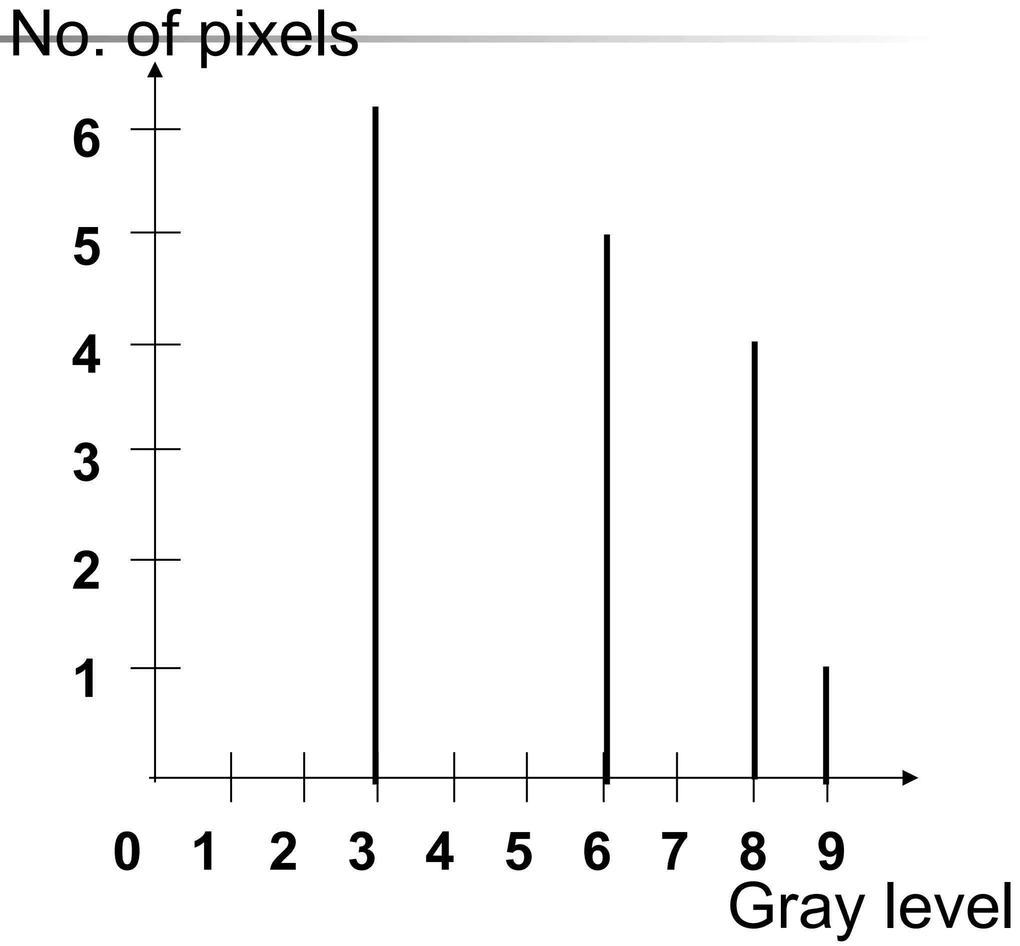
Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
s x 9	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

Example

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Gray scale = [0,9]



Histogram equalization

Note

- It is clearly seen that
  - Histogram equalization distributes the gray level to reach the maximum gray level (white) because the cumulative distribution function equals 1 when $0 \leq r \leq L-1$
  - If the cumulative numbers of gray levels are slightly different, they will be mapped to little different or same gray levels as we may have to approximate the processed gray level of the output image to integer number
  - Thus the discrete transformation function can't guarantee the one to one mapping relationship

Histogram Matching (Specification)

- Histogram equalization has a disadvantage which is that it can generate only one type of output image.
- With Histogram Specification, we can specify the shape of the histogram that we wish the output image to have.
- It doesn’t have to be a uniform histogram

Consider the continuous domain

Let $p_r(r)$ denote continuous probability density function of gray-level of input image, r

Let $p_z(z)$ denote desired (specified) continuous probability density function of gray-level of output image, z

Let s be a random variable with the property

$$s = T(r) = \int_0^r p_r(w)\,dw$$

Histogram equalization

Where w is a dummy variable of integration

Next, we define a random variable z with the property

$$g(z) = \int_0^z p_z(t)\,dt = s$$ ⇒ Histogram equalization

Where t is a dummy variable of integration

thus

$$s = T(r) = G(z)$$

Therefore, z must satisfy the condition

$$z = G^{-1}(s) = G^{-1}[T(r)]$$

Assume G^{-1} exists and satisfies the condition (a) and (b)

We can map an input gray level r to output gray level z

Procedure Conclusion

1. Obtain the transformation function T(r) by calculating the histogram equalization of the input image

$$s = T(r) = \int_0^r p_r(w)\,dw$$

2. Obtain the transformation function G(z) by calculating histogram equalization of the desired density function

$$G(z) = \int_0^z p_z(t)\,dt = s$$

Procedure Conclusion

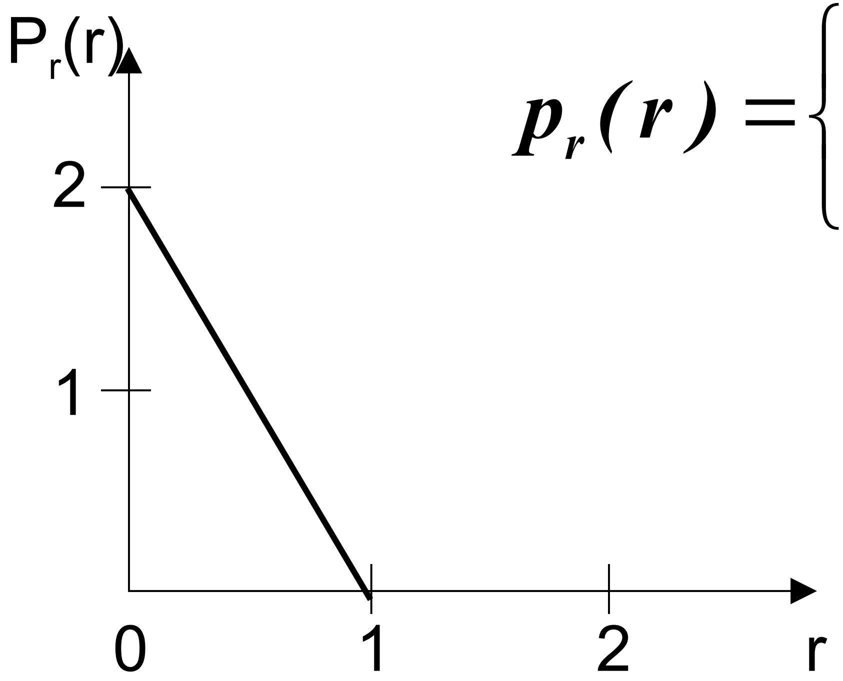
3. Obtain the inversed transformation function G^{-1}

$$z = G^{-1}(s) = G^{-1}[T(r)]$$

4. Obtain the output image by applying the processed gray-level from the inversed transformation function to all the pixels in the input image

Example

Assume an image has a gray level probability density function $p_r(r)$ as shown.

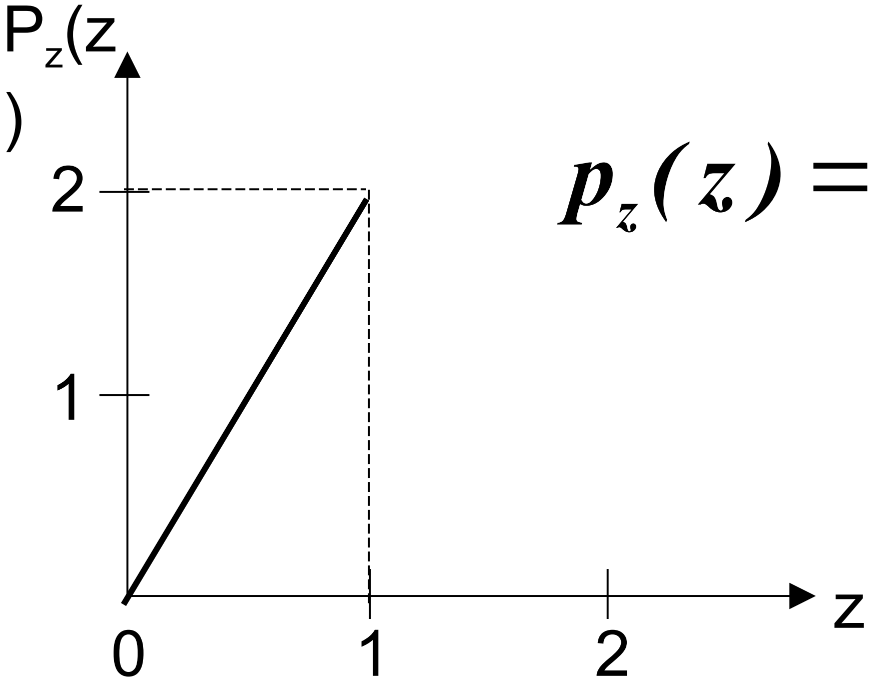


$$p_r(r) = \begin{cases} -2r+2 & ; 0 \leq r \leq 1 \\ 0 & ; \text{elsewhere} \end{cases}$$

$$\int_0^r p_r(w)\,dw = 1$$

Example

We would like to apply the histogram specification with the desired probability density function $p_z(z)$ as shown.

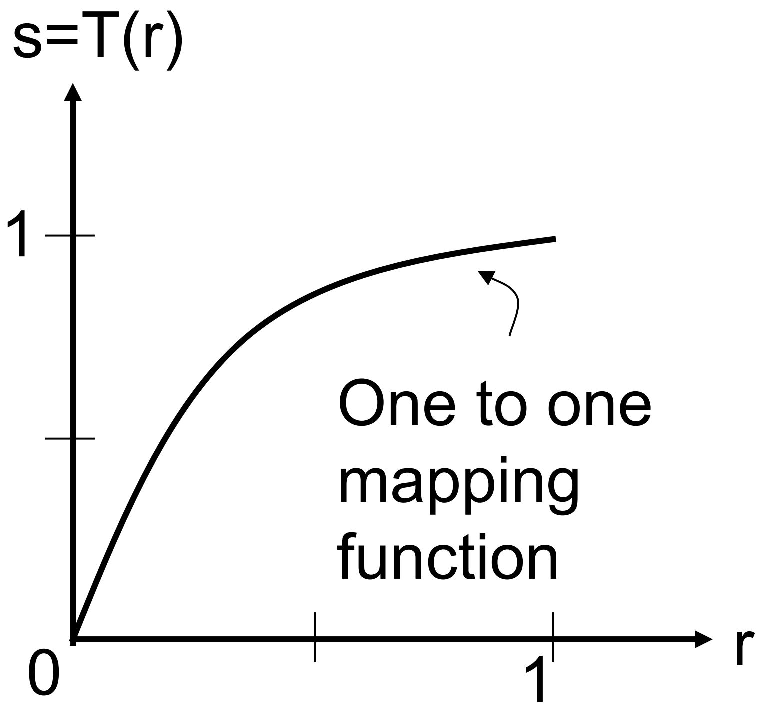


$$p_z(z) = \begin{cases} 2z & ;0 \le z \le 1 \\ 0 & ; \text{elsewhere} \end{cases}$$

$$\int_0^z p_z(w)\,dw = 1$$

Step 1:

Obtain the transformation function T(r)



$$s = T(r) = \int_0^r p_r(w)\,dw$$

$$= \int_0^r (-2w + 2)\,dw$$

$$= -w^2 + 2w \Big|_0^r$$

$$= -r^2 + 2r$$

Step 2:

Obtain the transformation function G(z)

$$G(z) = \int_0^z (2w)dw \qquad = z^2 \Big|_0^z \qquad = z^2$$

Step 3:

Obtain the inversed transformation function G^{-1}

$$G(z) = T(r)$$

$$z^2 = -r^2 + 2r$$

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

We can guarantee that $0 \leq z \leq 1$ when $0 \leq r \leq 1$

Discrete formulation

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0,1,2,\ldots,L-1$$

$$G(z_k) = \sum_{i=0}^{k} p_z(z_i) = s_k \qquad k = 0,1,2,\ldots,L-1$$

$$z_k = G^{-1}\left[T(r_k)\right]$$

$$= G^{-1}\left[s_k\right] \qquad k = 0,1,2,\ldots,L-1$$

Example

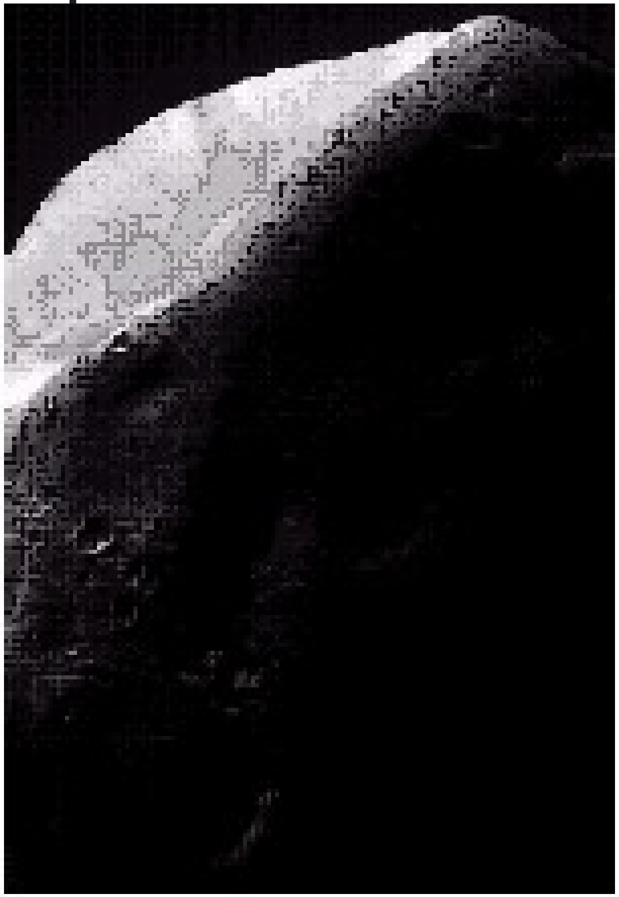


Image of Mars moon

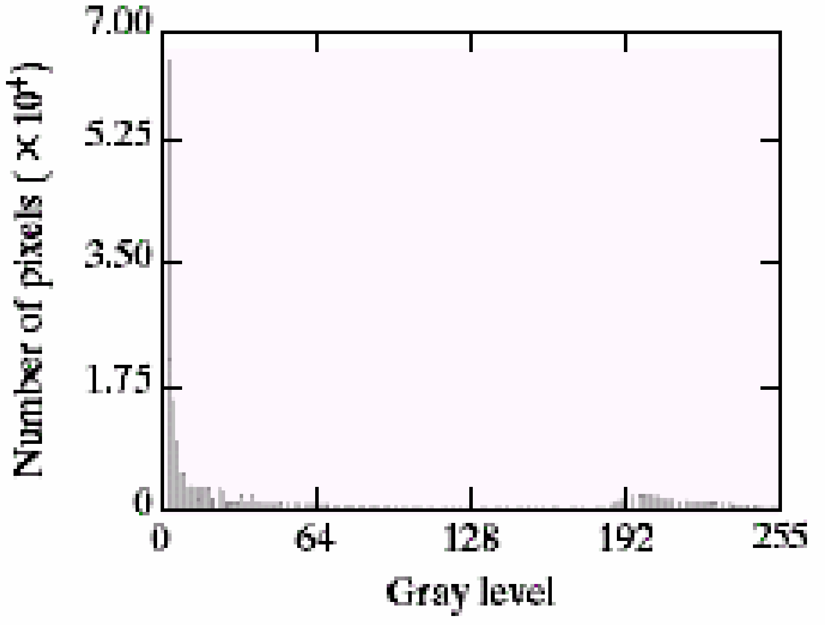
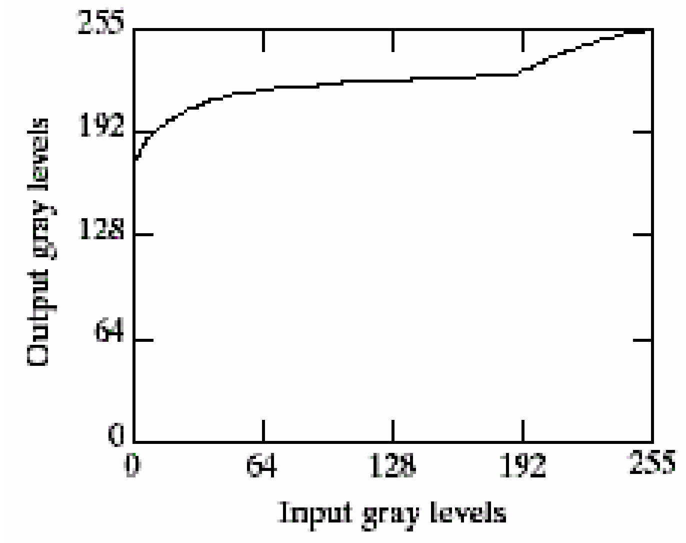
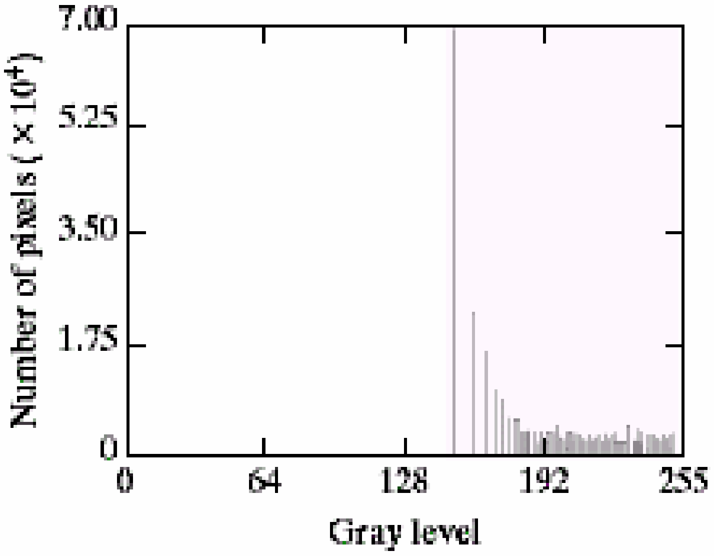


Image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in pixels in the dark end of the gray scale

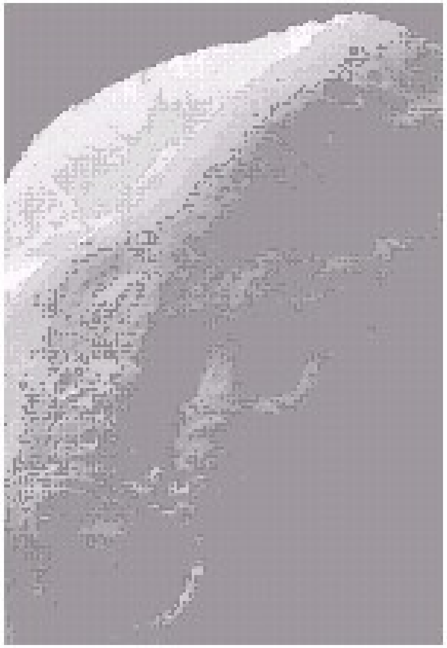
Image Equalization



Transformation function
for histogram equalization



Histogram of the result image



Result image
after histogram
equalization

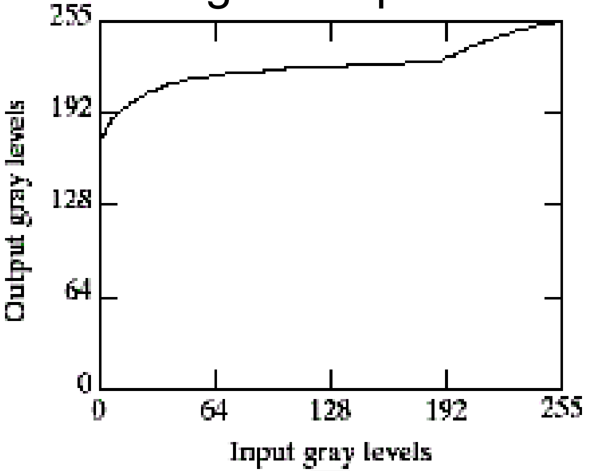
The histogram equalization doesn't make the result image look better than the original image. Consider the histogram of the result image, the net effect of this method is to map a very narrow interval of dark pixels into the upper end of the gray scale of the output image. As a consequence, the output image is light and has a washed-out appearance.

Solve the problem

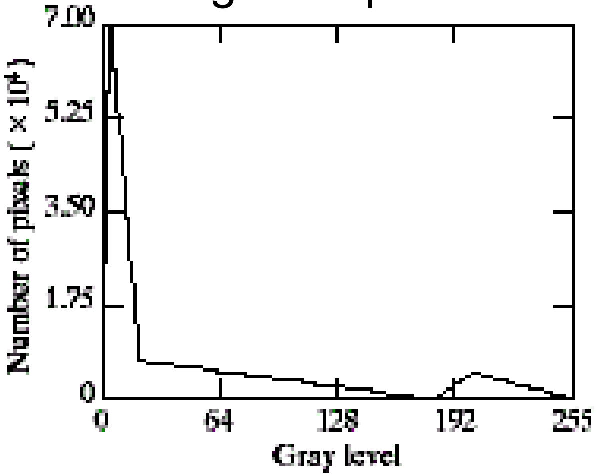
Since the problem with the transformation function of the histogram equalization was caused by a large concentration of pixels in the original image with levels near 0

a reasonable approach is to modify the histogram of that image so that it does not have this property

Histogram Equalization



Histogram Specification



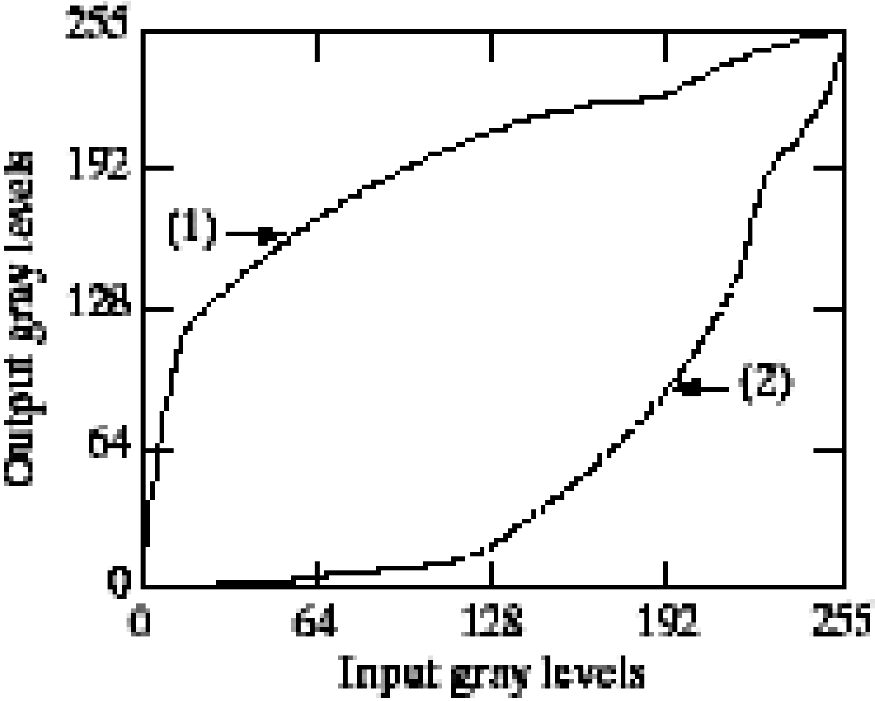
Histogram Specification

- (1) the transformation function G(z) obtained from

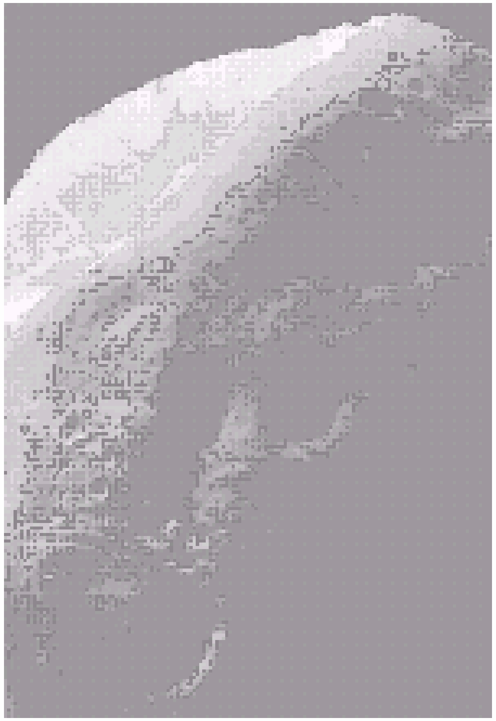
$$G(z_k) = \sum_{i=0}^{k} p_z(z_i) = s_k$$

$$k = 0,1,2,\ldots,L-1$$

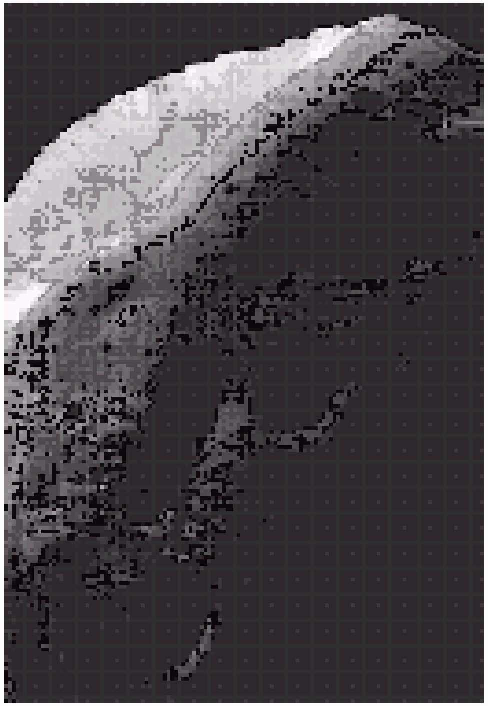
- (2) the inverse transformation $G^{-1}(s)$



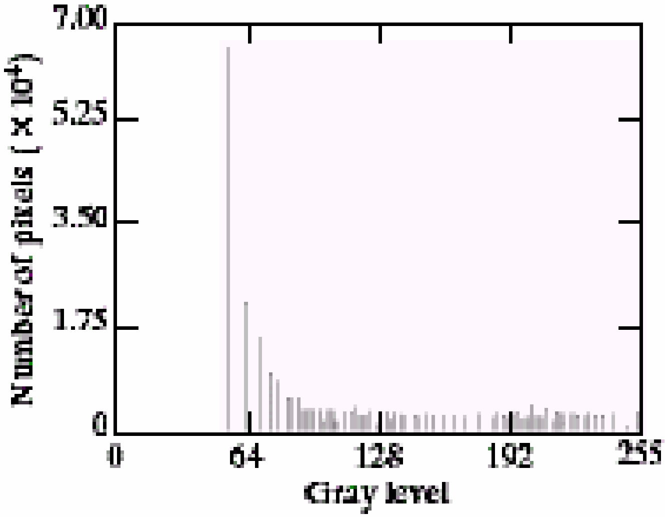
Result image and its histogram



Original image



After applied the histogram equalization



The output image's histogram

Notice that the output histogram's low end has shifted right toward the lighter region of the gray scale as desired.

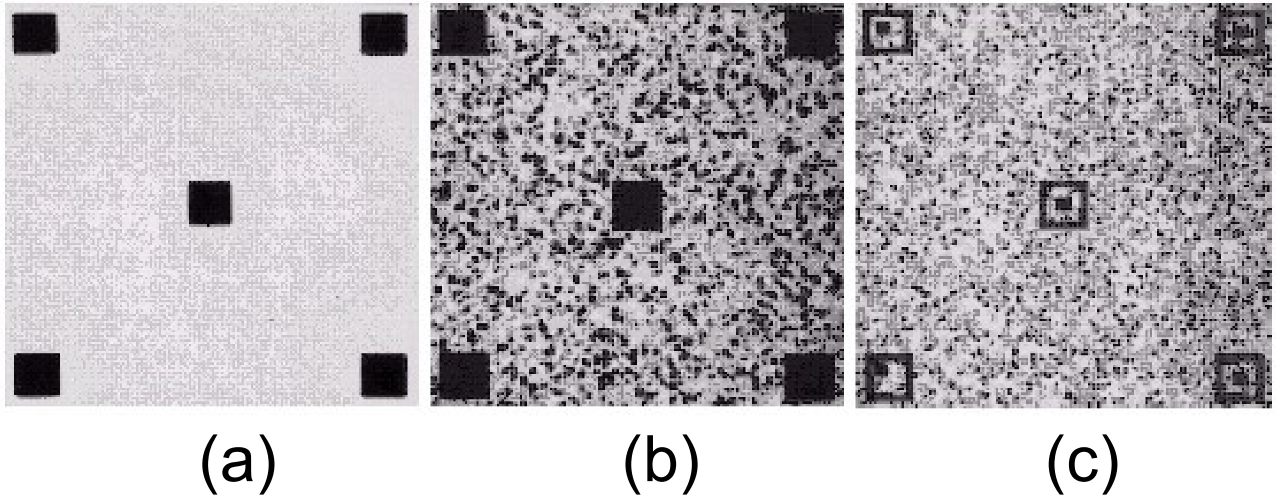
Note

- Histogram specification is a trial-and-error process
- There are no rules for specifying histograms, and one must resort to analysis on a case-by-case basis for any given enhancement task.

Note

- Histogram processing methods are global processing, in the sense that pixels are modified by a transformation function based on the gray-level content of an entire image.
- Sometimes, we may need to enhance details over small areas in an image, which is called a local enhancement.

Local Enhancement



a) Original image (slightly blurred to reduce noise)

b) global histogram equalization (enhance noise & slightly increase contrast but the construction is not changed)

c) local histogram equalization using 7x7 neighborhood (reveals the small squares inside larger ones of the original image.

- define a square or rectangular neighborhood and move the center of this area from pixel to pixel.
- at each location, the histogram of the points in the neighborhood is computed and either histogram equalization or histogram specification transformation function is obtained.
- another approach used to reduce computation is to utilize nonoverlapping regions, but it usually produces an undesirable checkerboard effect.

Explain the result in c)

- Basically, the original image consists of many small squares inside the larger dark ones.
- However, the small squares were too close in gray level to the larger ones, and their sizes were too small to influence global histogram equalization significantly.
- So, when we use the local enhancement technique, it reveals the small areas.
- Note also the finer noise texture is resulted by the local processing using relatively small neighborhoods.

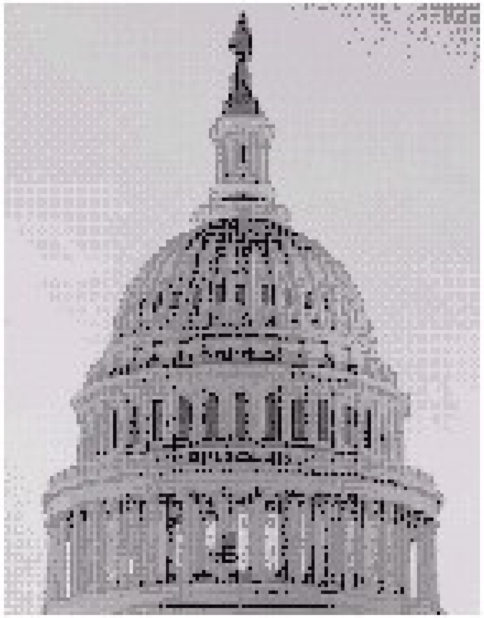
Enhancement using Arithmetic/ Logic Operations

- Arithmetic/Logic operations perform on pixel by pixel basis between two or more images
- except NOT operation which perform only on a single image

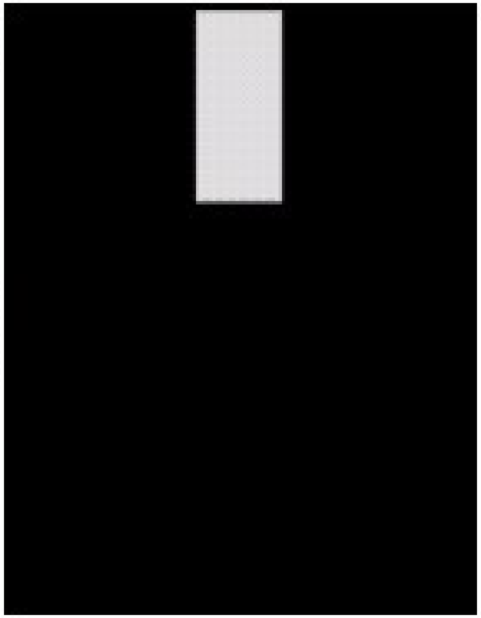
Logic Operations

- Logic operation performs on gray-level images, the pixel values are processed as binary numbers
- light represents a binary 1, and dark represents a binary 0
- NOT operation = negative transformation

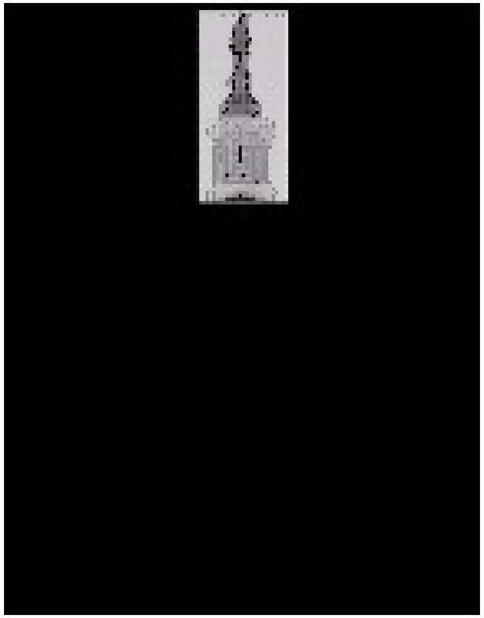
Example of AND Operation



original image

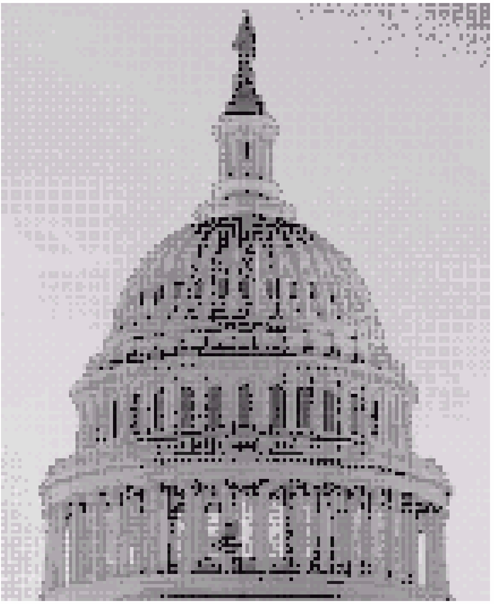


AND image mask

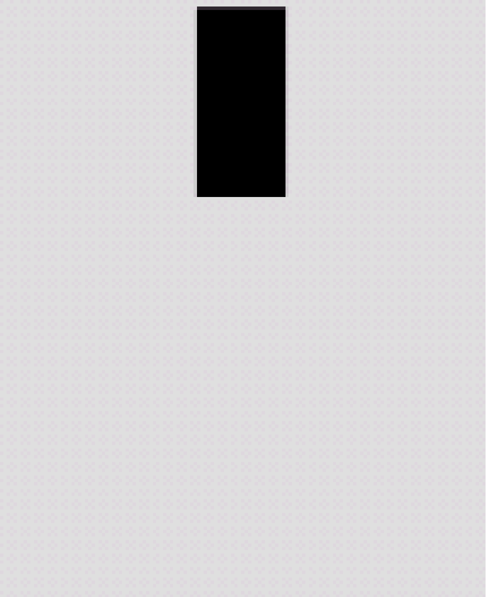


result of AND operation

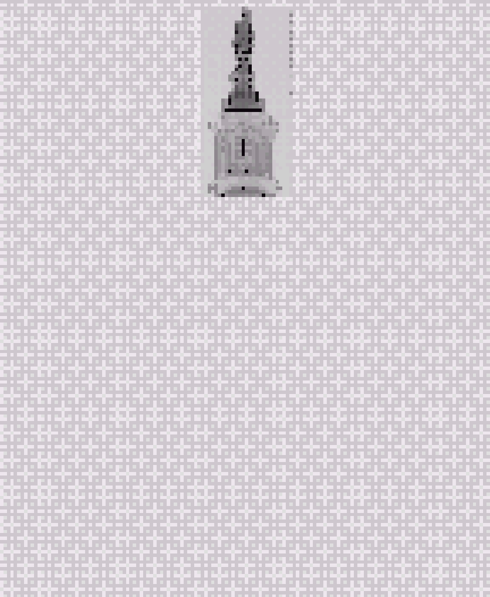
Example of OR Operation



original image



OR image mask



result of OR operation

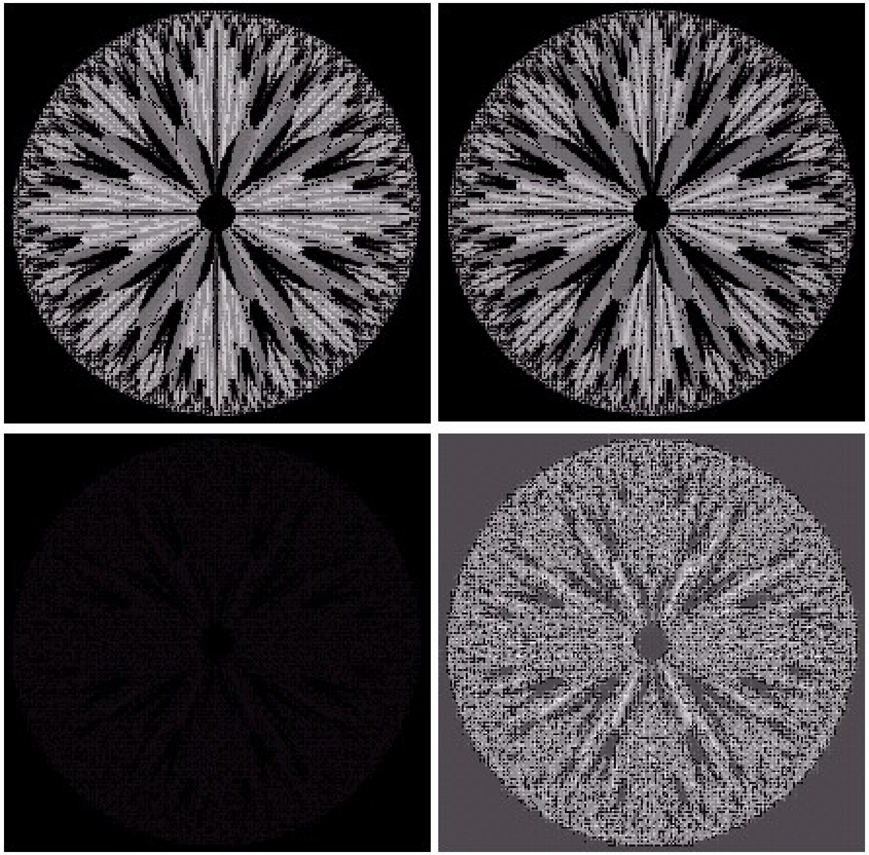
Image Subtraction

$$g(x,y) = f(x,y) - h(x,y)$$

- enhancement of the differences between images

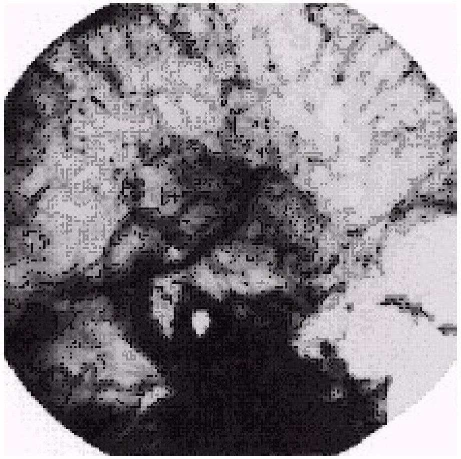
a	b
c	d

Image Subtraction

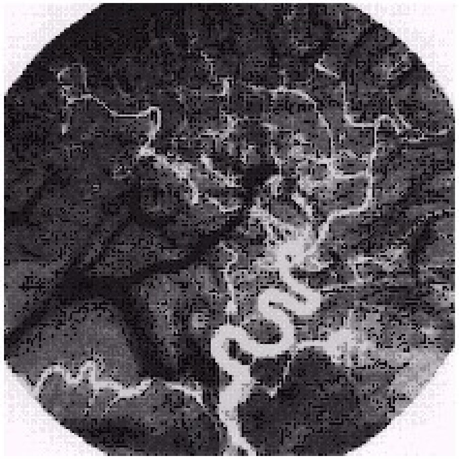


- a). original fractal image
- b). result of setting the four lower-order bit planes to zero
  - refer to the bit-plane slicing
  - the higher planes contribute significant detail
  - the lower planes contribute more to fine detail
  - image b). is nearly identical visually to image a), with a very slightly drop in overall contrast due to less variability of the gray-level values in the image.
- c). difference between a). and b). (nearly black)
- d). histogram equalization of c). (perform contrast stretching transformation)

Mask mode radiography



mask image



an image (taken after injection of a contrast medium (iodine) into the bloodstream) with mask subtracted out.

Note:

• the background is dark because it doesn't change much in both images.

• the difference area is bright because it has a big change

- h(x,y) is the mask, an X-ray image of a region of a patient's body captured by an intensified TV camera (instead of traditional X-ray film) located opposite an X-ray source
- f(x,y) is an X-ray image taken after injection a contrast medium into the patient's bloodstream
- images are captured at TV rates, so the doctor can see how the medium propagates through the various arteries in the area being observed (the effect of subtraction) in a movie showing mode.

Note

- We may have to adjust the gray-scale of the subtracted image to be [0, 255] (if 8-bit is used)
  - first, find the minimum gray value of the subtracted image
  - second, find the maximum gray value of the subtracted image
  - set the minimum value to be zero and the maximum to be 255
  - while the rest are adjusted according to the interval [0, 255], by timing each value with 255/max
- Subtraction is also used in segmentation of moving pictures to track the changes
  - after subtract the sequenced images, what is left should be the moving elements in the image, plus noise

Image Averaging

- consider a noisy image $g(x,y)$ formed by the addition of noise $\eta(x,y)$ to an original image $f(x,y)$

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

Image Averaging

- if noise has zero mean and be uncorrelated then it can be shown that if

$\bar{g}(x, y)$ = image formed by averaging K different noisy images

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

Image Averaging

- then

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

$\sigma^2{}_{\bar{g}(x,y)}, \sigma^2{}_{\eta(x,y)}$ = variances of $\bar{g}$ and η

if K increase, it indicates that the variability (noise) of the pixel at each location (x,y) decreases.

Image Averaging

- thus

$$\boxed{E\{\overline{g}(x,y)\} = f(x,y)}$$

$E\{\overline{g}(x,y)\}$ = expected value of $\overline{g}$ (output after averaging)

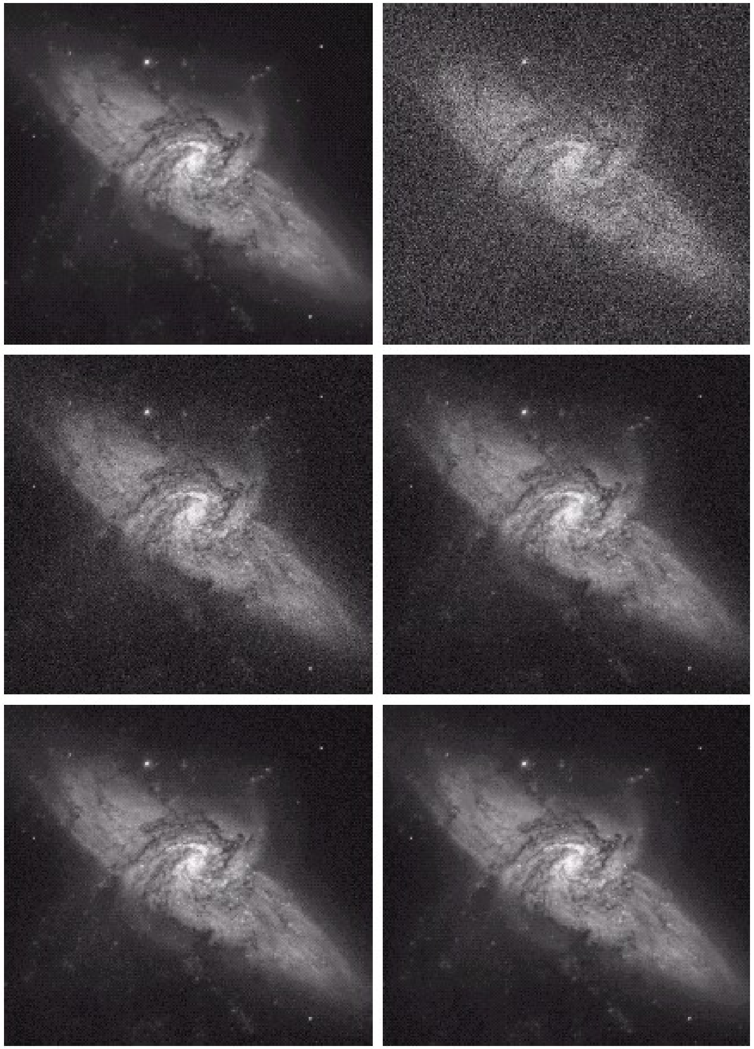
= original image f(x,y)

Image Averaging

- Note: the images $g_i(x,y)$ (noisy images) must be registered (aligned) in order to avoid the introduction of blurring and other artifacts in the output image.

a	b
c	d
e	f

Example



- a) original image
- b) image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels.
- c). -f). results of averaging K = 8, 16, 64 and 128 noisy images

Spatial Filtering

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

- use filter (can also be called as mask/kernel/template or window)
- the values in a filter subimage are referred to as coefficients, rather than pixel.
- our focus will be on masks of odd sizes, e.g. 3x3, 5x5,…

Spatial Filtering Process

- simply move the filter mask from point to point in an image.
- at each point (x,y), the response of the filter at that point is calculated using a predefined relationship.

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn}$$

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

Linear Filtering

- Linear Filtering of an image f of size MxN filter mask of size mxn is given by the expression

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

where a = (m-1)/2 and b = (n-1)/2

To generate a complete filtered image this equation must be applied for x = 0, 1, 2, … , M-1 and y = 0, 1, 2, … , N-1

Smoothing Spatial Filters

- used for blurring and for noise reduction
- blurring is used in preprocessing steps, such as
  - removal of small details from an image prior to object extraction
  - bridging of small gaps in lines or curves
- noise reduction can be accomplished by blurring with a linear filter and also by a nonlinear filter

Smoothing Linear Filters

- output is simply the average of the pixels contained in the neighborhood of the filter mask.
- called averaging filters or lowpass filters.

Smoothing Linear Filters

- replacing the value of every pixel in an image by the average of the gray levels in the neighborhood will reduce the "sharp" transitions in gray levels.
- sharp transitions
  - random noise in the image
  - edges of objects in the image
- thus, smoothing can reduce noises (desirable) and blur edges (undesirable)

3x3 Smoothing Linear Filters

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

box filter

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

weighted average

the center is the most important and other pixels are inversely weighted as a function of their distance from the center of the mask

Weighted average filter

- the basic strategy behind weighting the center point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply an attempt to reduce blurring in the smoothing process.

General form : smoothing mask

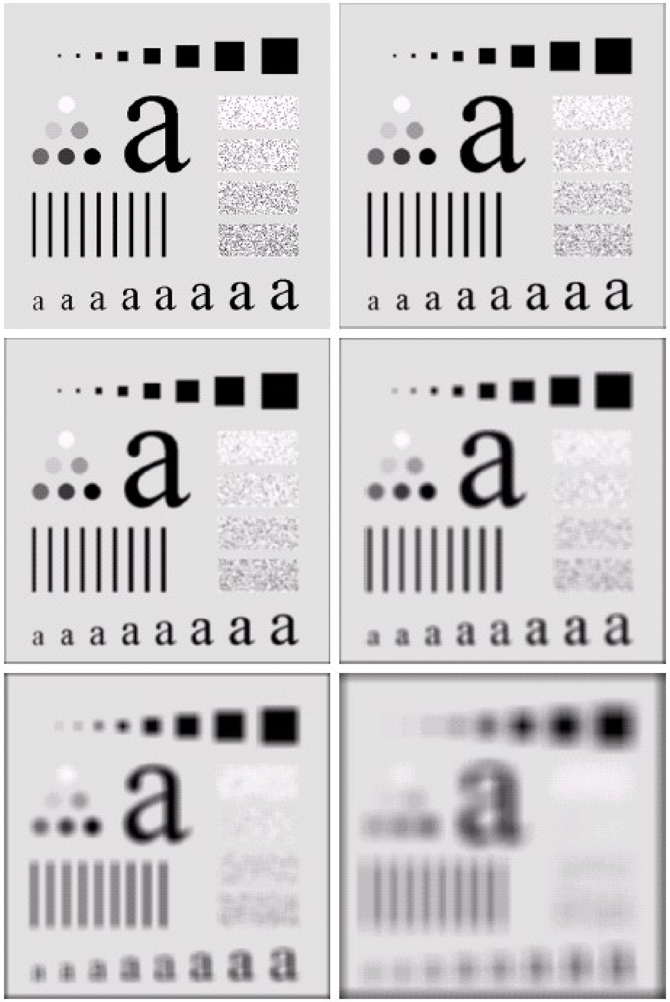
- filter of size mxn (m and n odd)

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

summation of all coefficient of the mask

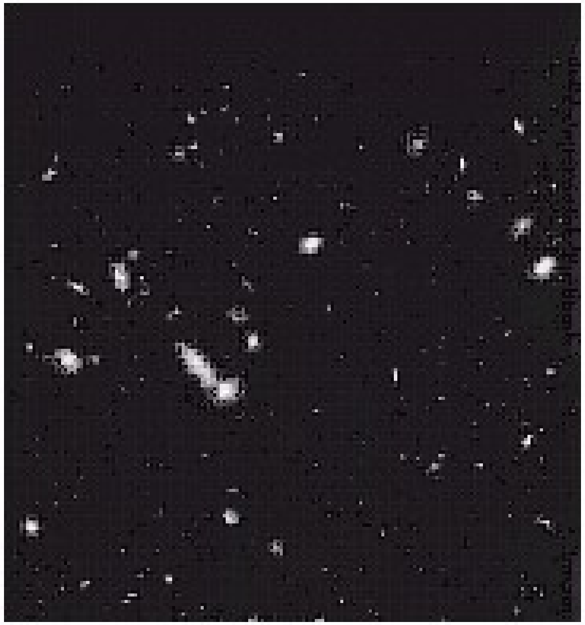
a	b
c	d
e	f

Example

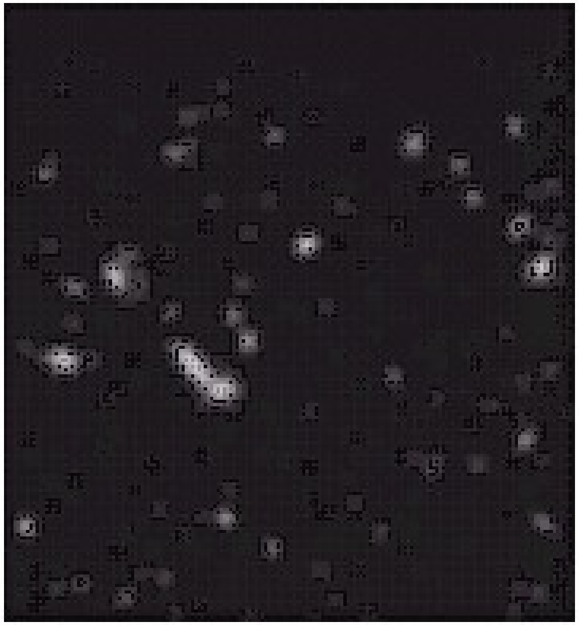


- a). original image 500x500 pixel
- b). - f). results of smoothing with square averaging filter masks of size n = 3, 5, 9, 15 and 35, respectively.
- Note:
  - big mask is used to eliminate small objects from an image.
  - the size of the mask establishes the relative size of the objects that will be blended with the background.

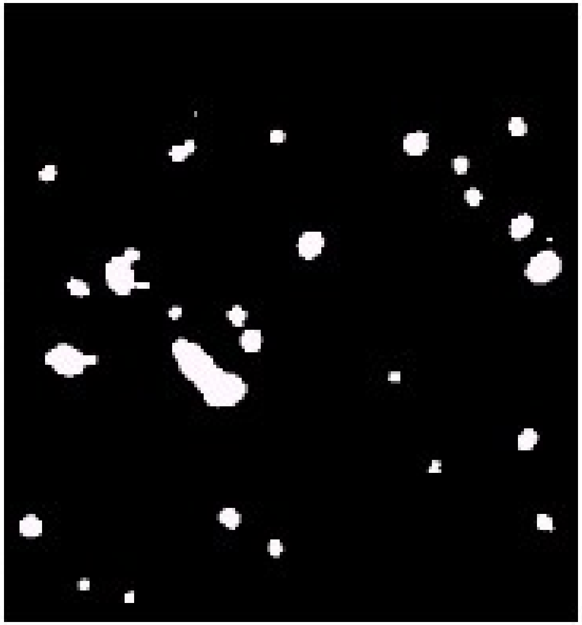
Example



original image



result after smoothing with 15x15 averaging mask



result of thresholding

we can see that the result after smoothing and thresholding, the remains are the largest and brightest objects in the image.

Order-Statistics Filters (Nonlinear Filters)

- the response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter
- example
  - median filter : R = median$\{z_k \mid k = 1,2,\ldots,n \times n\}$
  - max filter : R = max$\{z_k \mid k = 1,2,\ldots,n \times n\}$
  - min filter : R = min$\{z_k \mid k = 1,2,\ldots,n \times n\}$
- note: n x nis the size of the mask

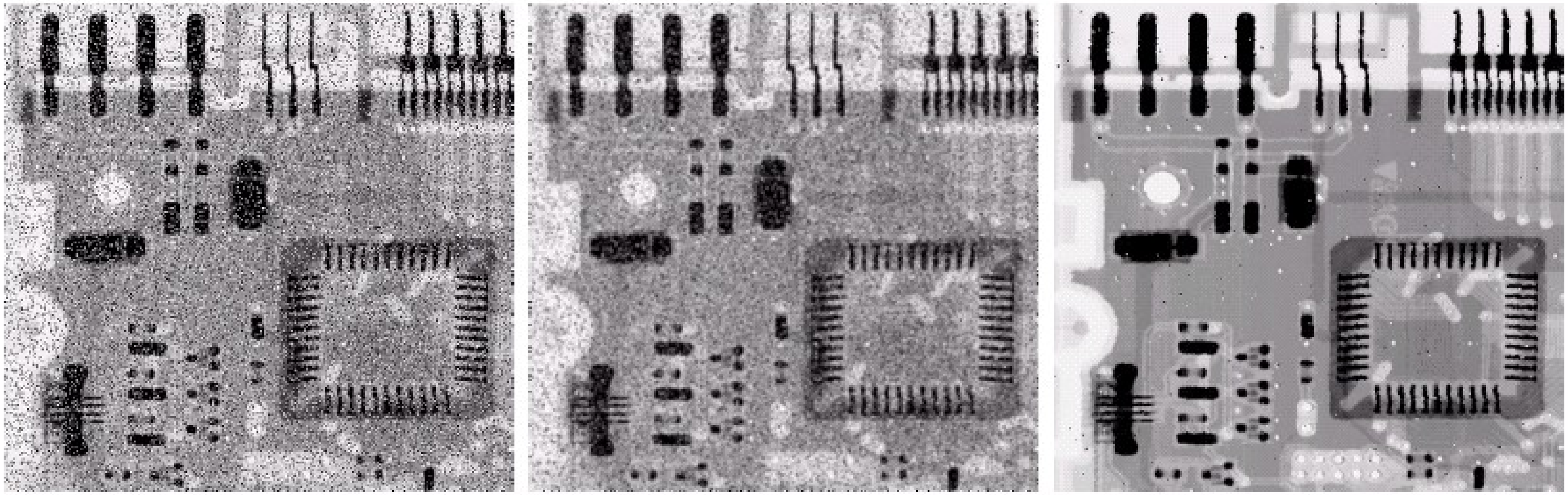
Median Filters

- replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median)
- quite popular because for certain types of random noise (**impulse noise □ salt and pepper noise**) , they **provide excellent noise-reduction capabilities**, with considering **less blurring than linear smoothing filters of similar size.**

Median Filters

- forces the points with distinct gray levels to be more like their neighbors.
- isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2/2$ (one-half the filter area), are eliminated by an n x n median filter.
- eliminated = forced to have the value equal the median intensity of the neighbors.
- larger clusters are affected considerably less

Example : Median Filters



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Sharpening Spatial Filters

- to highlight fine detail in an image
- or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.

Blurring vs. Sharpening

- as we know that blurring can be done in spatial domain by pixel averaging in a neighbors
- since averaging is analogous to integration
- thus, we can guess that the sharpening must be accomplished by spatial differentiation.

Derivative operator

- the strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.
- thus, image differentiation
  - enhances edges and other discontinuities (noise)
  - deemphasizes area with slowly varying gray-level values.

First-order derivative

- a basic definition of the first-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

Second-order derivative

- similarly, we define the second-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

First and Second-order derivative of f(x,y)

- when we consider an image function of two variables, f(x,y), at which time we will dealing with partial derivatives along the two spatial axes.

Gradient operator

$$\nabla \mathrm{f} = \frac{\partial f(x,y)}{\partial x \partial y} = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}$$

Laplacian operator

(linear operator)

$$\nabla^2 f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

Discrete Form of Laplacian

from

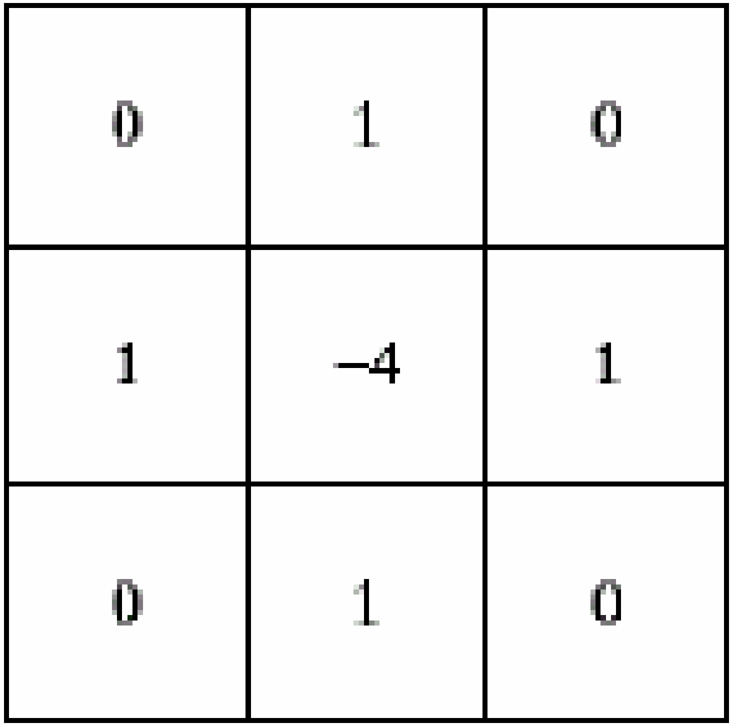
$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

yield,

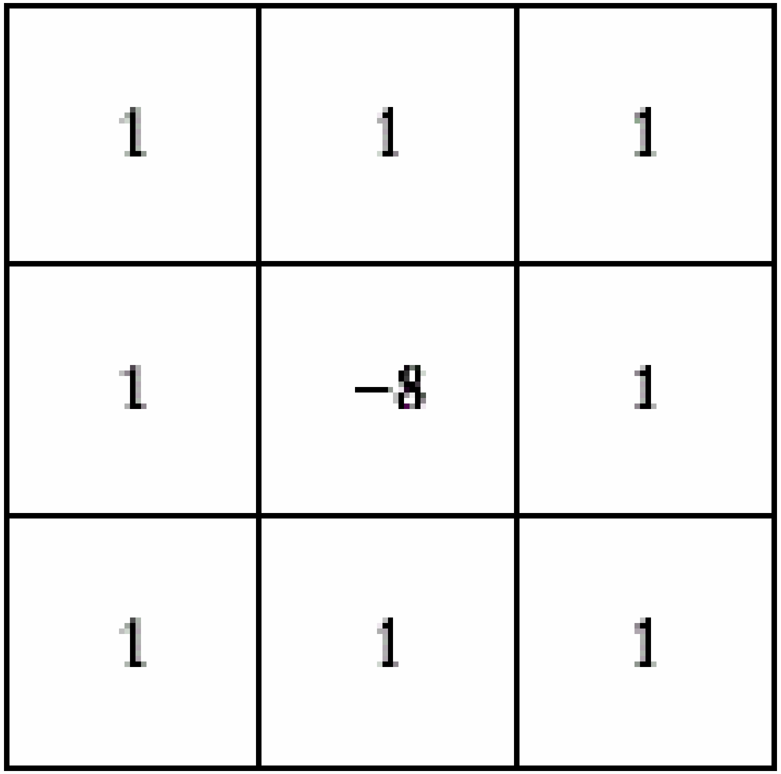
$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)]$$

Result Laplacian mask



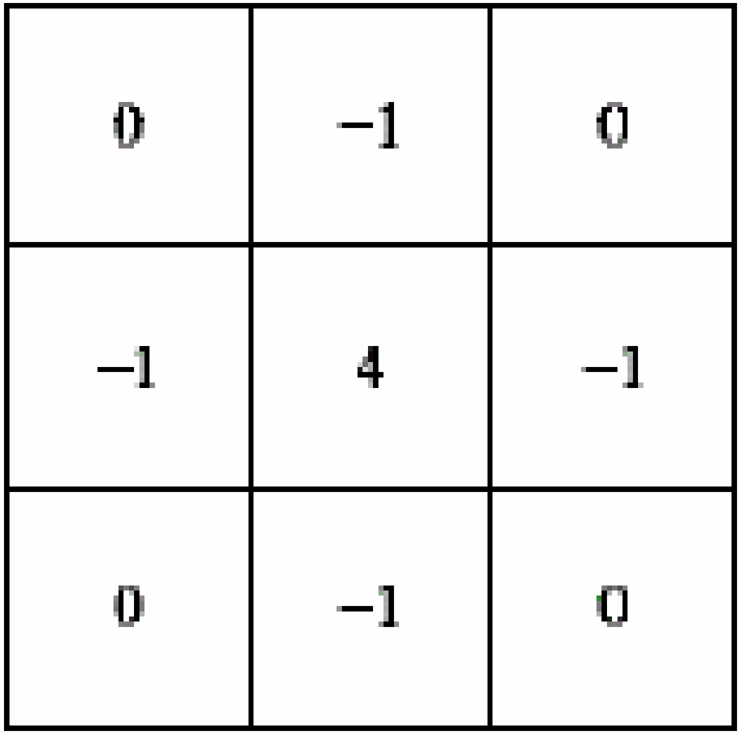
0	1	0
1	−4	1
0	1	0

Laplacian mask implemented an extension of diagonal neighbors

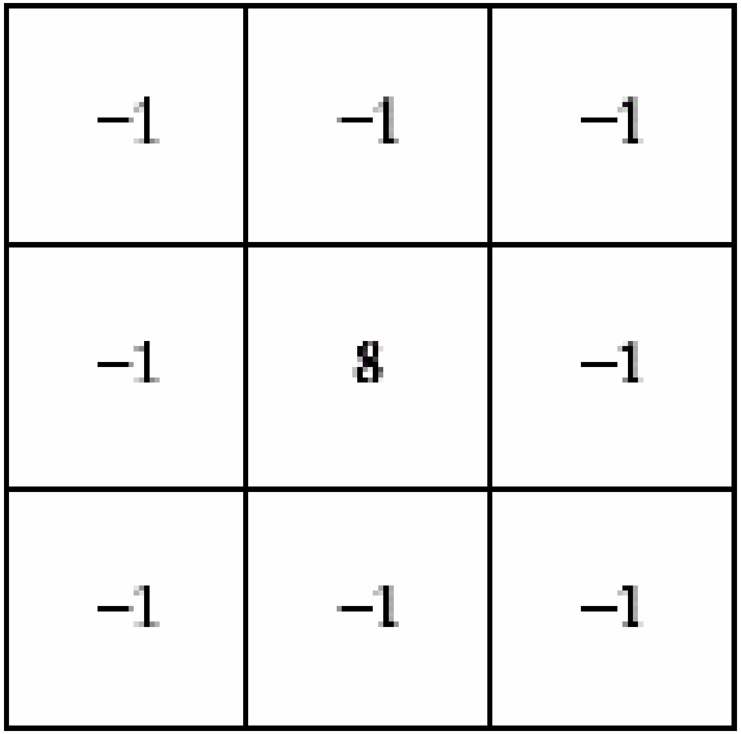


1	1	1
1	−8	1
1	1	1

Other implementation of Laplacian masks



0	−1	0
−1	4	−1
0	−1	0



−1	−1	−1
−1	8	−1
−1	−1	−1

give the same result, but we have to keep in mind that when combining (add / subtract) a Laplacian-filtered image with another image.

Effect of Laplacian Operator

- as it is a derivative operator,
  - it highlights gray-level discontinuities in an image
  - it deemphasizes regions with slowly varying gray levels
- tends to produce images that have
  - grayish edge lines and other discontinuities, all superimposed on a dark,
  - featureless background.

Correct the effect of featureless background

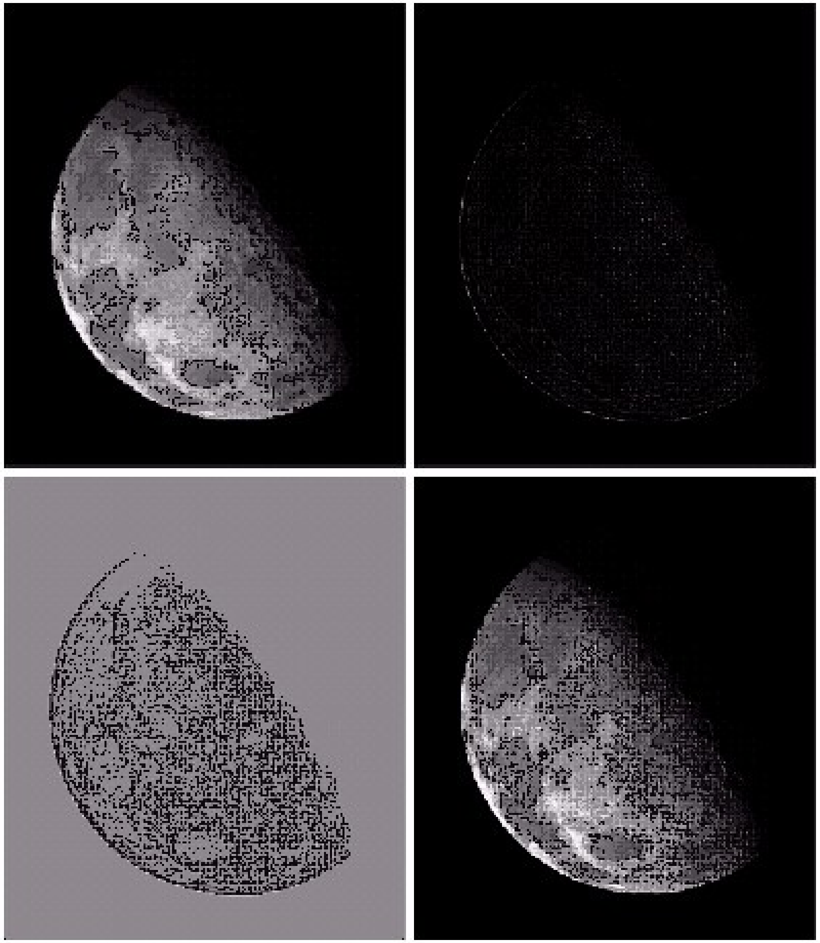
- easily by adding the original and Laplacian image.
- be careful with the Laplacian filter used

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$$

if the center coefficient of the Laplacian mask is negative

if the center coefficient of the Laplacian mask is positive

Example



- a). image of the North pole of the moon
- b). Laplacian-filtered image with

1	1	1
1	-8	1
1	1	1

- c). Laplacian image scaled for display purposes
- d). image enhanced by addition with original image

Mask of Laplacian + addition

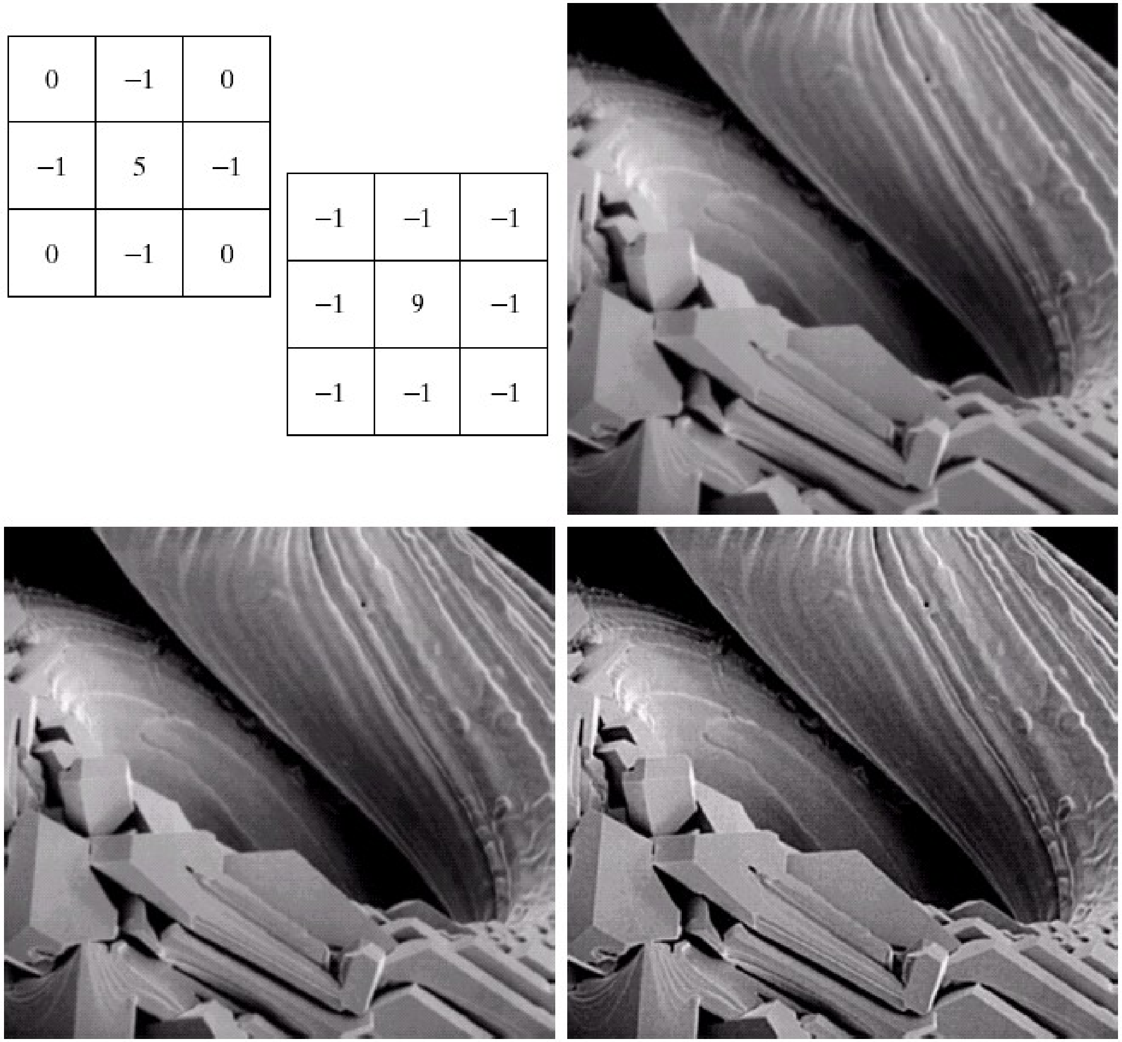
- to simply the computation, we can create a mask which do both operations, Laplacian Filter and Addition the original image.

Mask of Laplacian + addition

$$
\begin{aligned}
g(x,y) &= f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\qquad + f(x,y+1) + f(x,y-1) - 4f(x,y)] \\
&= 5f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\qquad + f(x,y+1) + f(x,y-1)]
\end{aligned}
$$

0	-1	0
-1	5	-1
0	-1	0

Example



0	–1	0
–1	5	–1
0	–1	0

–1	–1	–1
–1	9	–1
–1	–1	–1

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Note

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 9 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 8 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

Unsharp masking

$$f_s(x,y) = f(x,y) - \bar{f}(x,y)$$

sharpened image = original image – blurred image

- to subtract a blurred version of an image produces sharpening output image.

High-boost filtering

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y)$$

$$\begin{aligned} f_{hb}(x,y) &= (A-1)f(x,y) + f(x,y) - \bar{f}(x,y) \\ &= (A-1)f(x,y) - f_s(x,y) \end{aligned}$$

- generalized form of Unsharp masking
- $A \geq 1$

High-boost filtering

$$f_{hb}(x,y) = (A-1)f(x,y) - f_s(x,y)$$

- if we use Laplacian filter to create sharpen image $f_s(x,y)$ with addition of original image

$$f_s(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$$

High-boost filtering

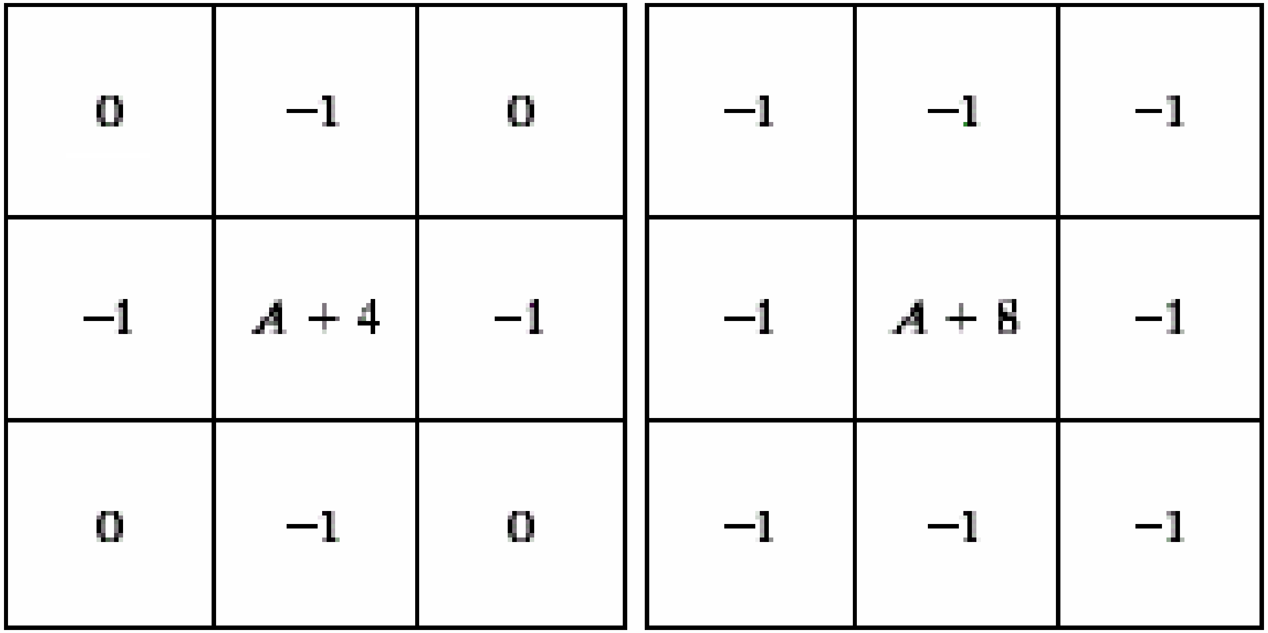
- yields

if the center coefficient of the Laplacian mask is negative

$$f_{hb}(x,y) = \begin{cases} Af(x,y) - \nabla^2 f(x,y) \\ Af(x,y) + \nabla^2 f(x,y) \end{cases}$$

if the center coefficient of the Laplacian mask is positive

High-boost Masks



0	−1	0
−1	$A+4$	−1
0	−1	0

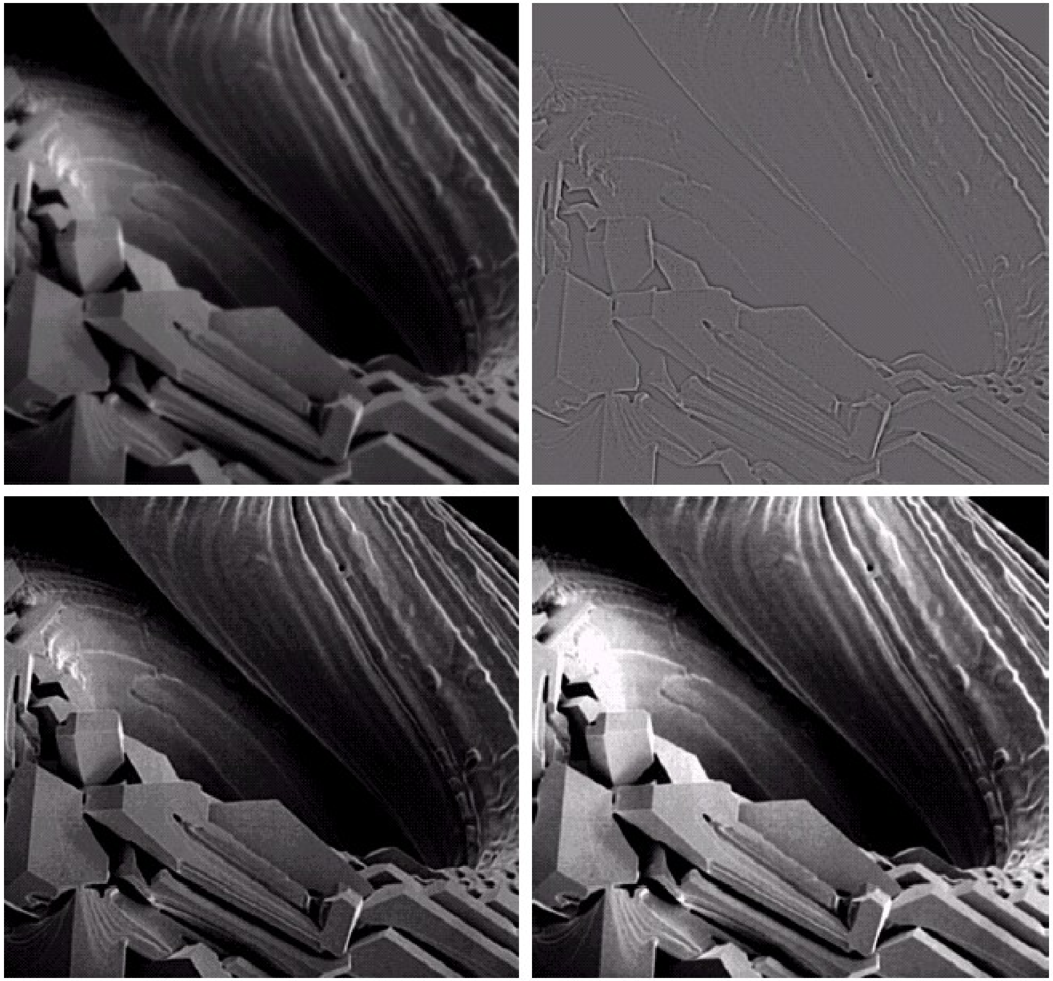
−1	−1	−1
−1	$A+8$	−1
−1	−1	−1

- $A \geq 1$
- if $A = 1$, it becomes “standard” Laplacian sharpening

Example

a b
c d

FIGURE 3.43 (a) Same as Fig. 3.41(c), but darker. (a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$. (c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.



Gradient Operator

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- first derivatives are implemented using the **magnitude of the gradient**.

$$\nabla f = mag(\nabla \mathrm{f}) = [G_x^2 + G_y^2]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

the magnitude becomes nonlinear

commonly approx.

$$\nabla f \approx |G_x| + |G_y|$$

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- simplest approximation, 2x2

$$G_x = (z_8 - z_5) \quad \text{and} \quad G_y = (z_6 - z_5)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_8 - z_5)^2 + (z_6 - z_5)^2]^{1/2}$$

$$\nabla f \approx |z_8 - z_5| + |z_6 - z_5|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Gradient Mask

- Roberts cross-gradient operators, 2x2

$$G_x = (z_9 - z_5) \quad \text{and} \quad G_y = (z_8 - z_6)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

−1	0
0	1

0	−1
1	0

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Sobel operators, 3x3

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$\nabla f \approx |G_x| + |G_y|$$

the weight value 2 is to achieve smoothing by giving more important to the center point

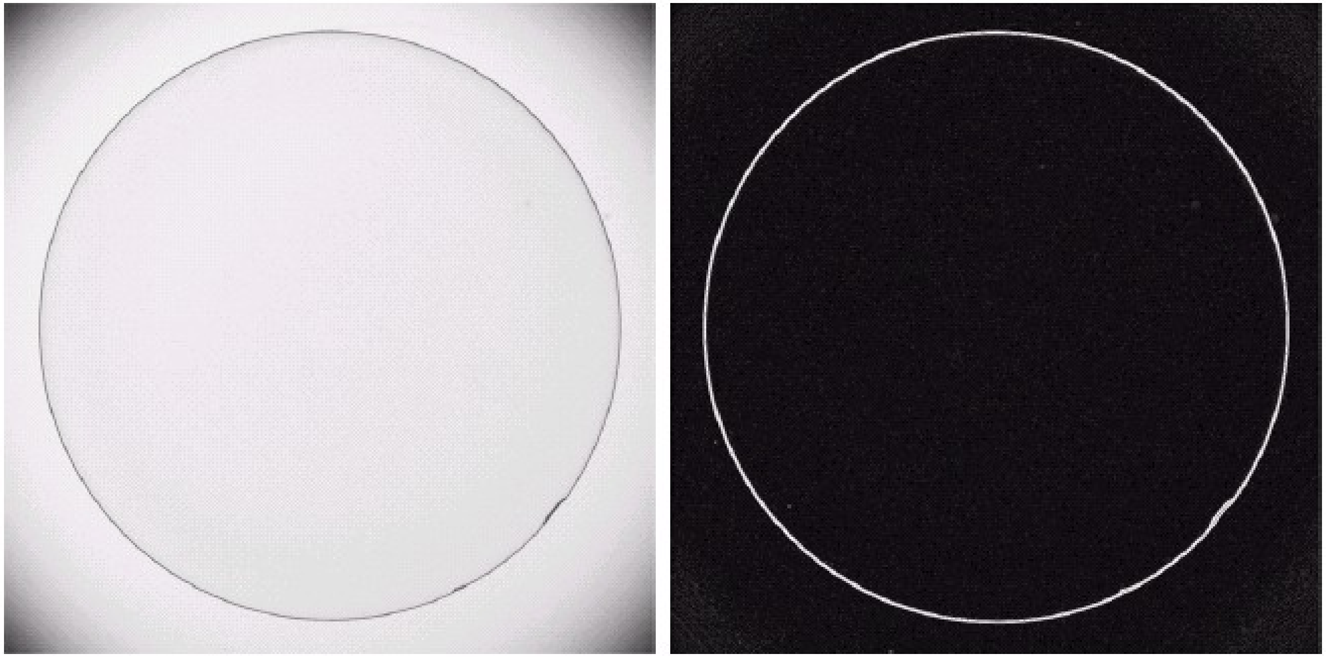
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Note

- the summation of coefficients in all masks equals 0, indicating that they would give a response of 0 in an area of constant gray level.

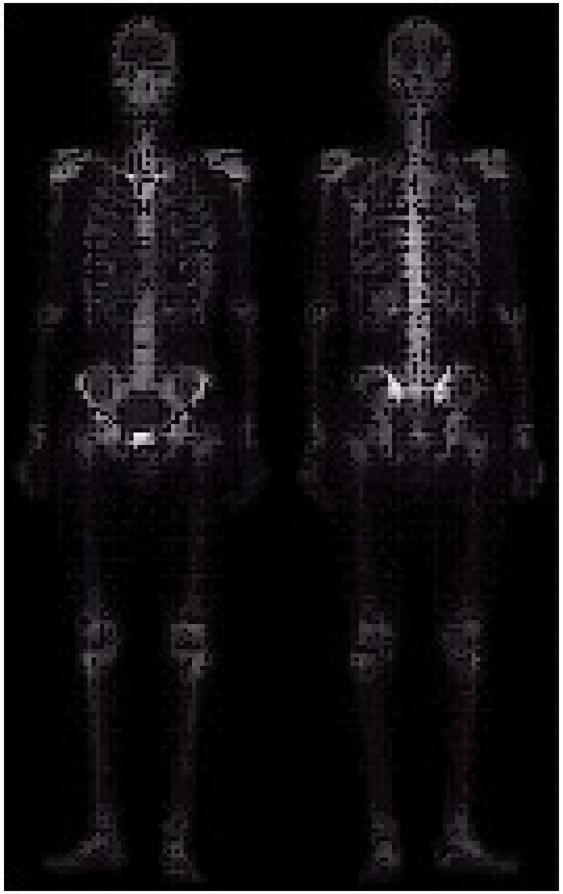
Example



a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Example of Combining Spatial Enhancement Methods

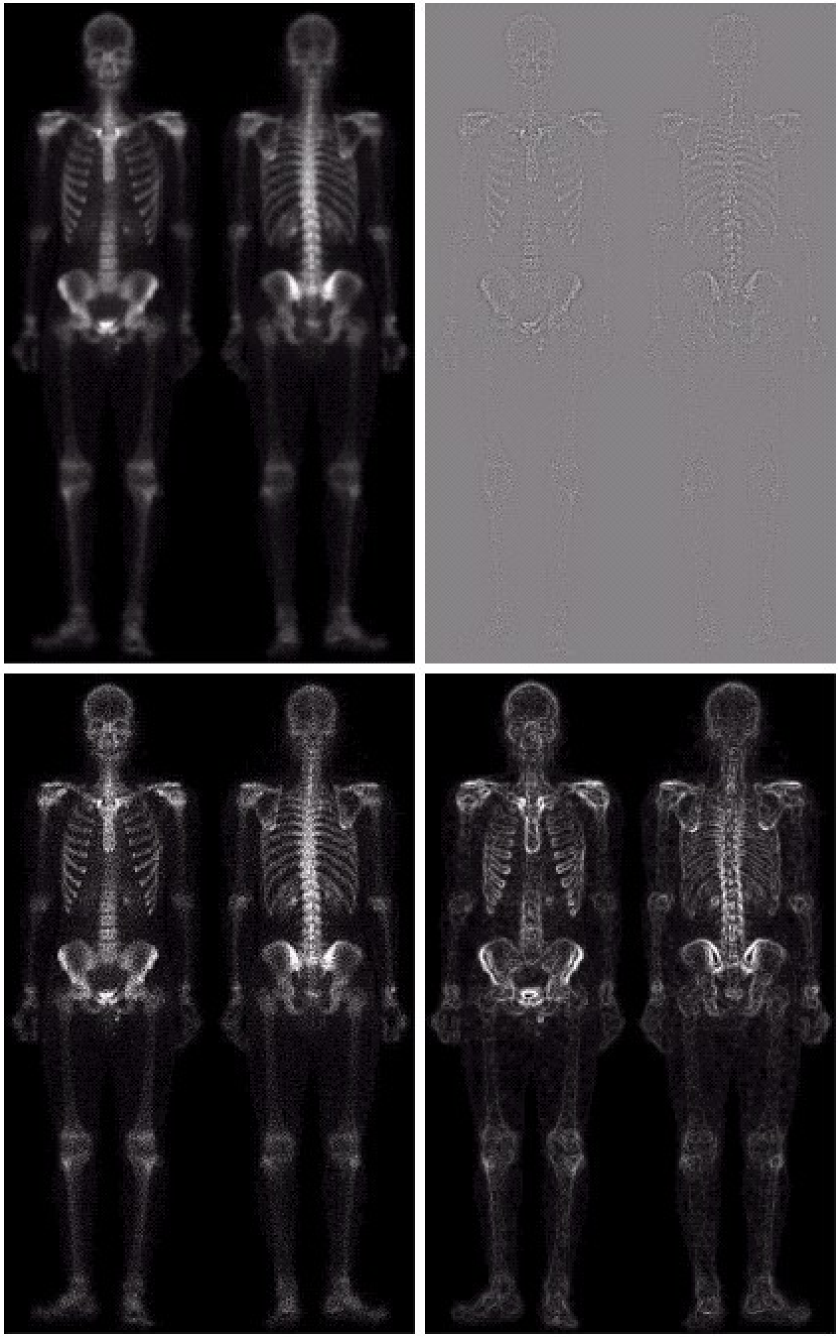


- want to sharpen the original image and bring out more skeletal detail.
- problems: narrow dynamic range of gray level and high noise content makes the image difficult to enhance

Example of Combining Spatial Enhancement Methods

- solve :

1. Laplacian to highlight fine detail
2. gradient to enhance prominent edges
3. gray-level transformation to increase the dynamic range of gray levels



a	b
c	d

FIGURE 3.46 (a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).

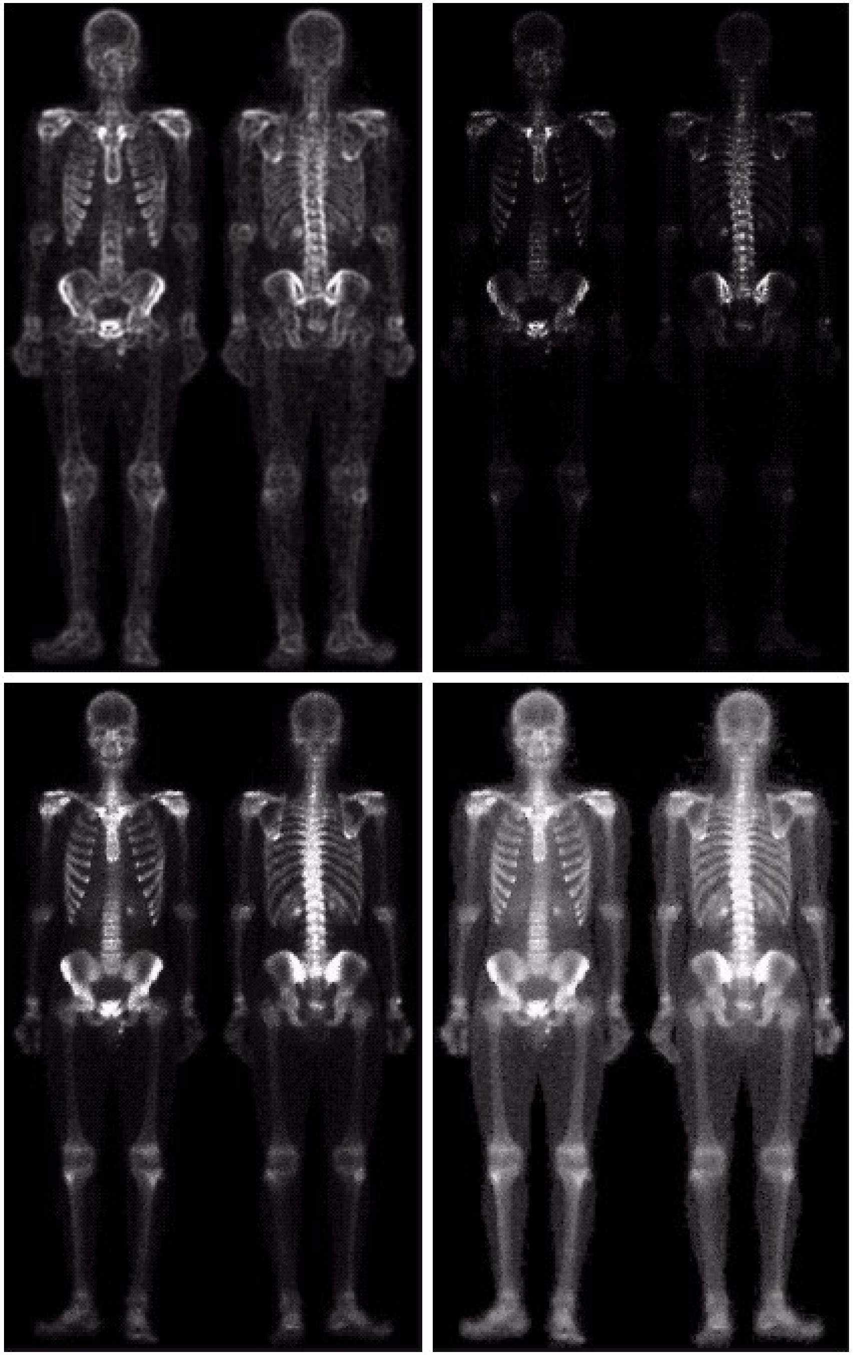


FIGURE 3.46 *(Continued)* (e) Sobel image smoothed with a 5 × 5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)