Digital Image Processing

Chapter 3:
Image Enhancement in the Spatial Domain

Principle Objective of Enhancement

- Process an image so that the result will be more suitable than the original image for a specific application.
- Techniques are problem oriented.
- A method which is quite useful for enhancing an image may not necessarily be the best approach for enhancing another images
- No general theory on image enhancement exists.

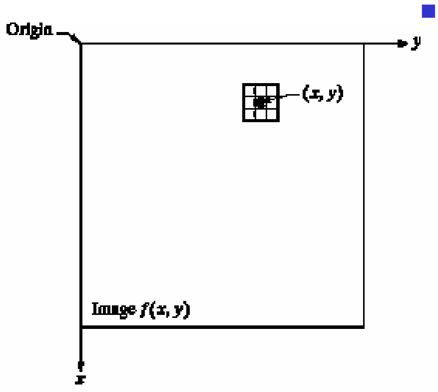
2 domains

- Spatial Domain (image plane):
  - Techniques are based on direct manipulation of pixels in an image.
    - Gray level transformations.
    - Histogram processing.
    - Arithmetic/Logic operations.
    - Filtration techniques.
- Frequency Domain :
  - Techniques are based on modifying the Fourier transform of an image

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

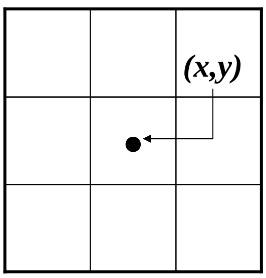


- Procedures that operate directly on pixels.

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

  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

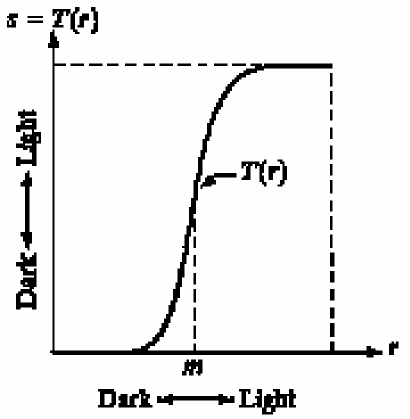
Point Processing

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

$$s = T(r)$$

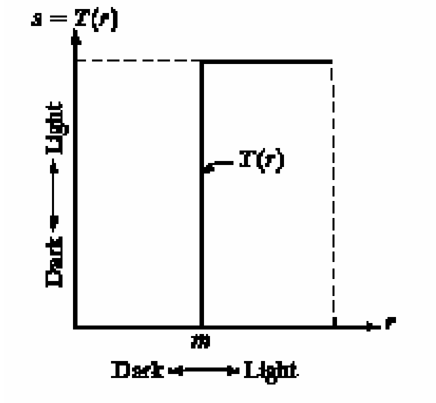
- Where
  - r = gray level of $f(x,y)$
  - s = gray level of $g(x,y)$

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

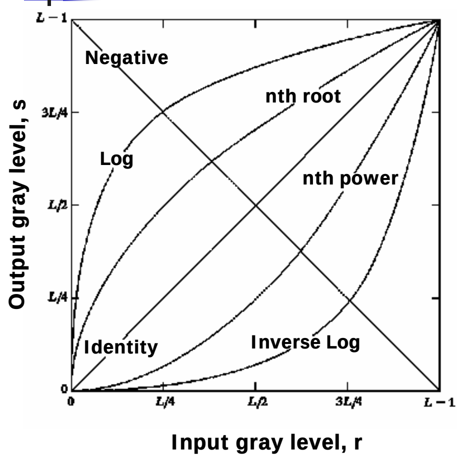


- Produce a two-level (binary) image

Mask Processing or Filter

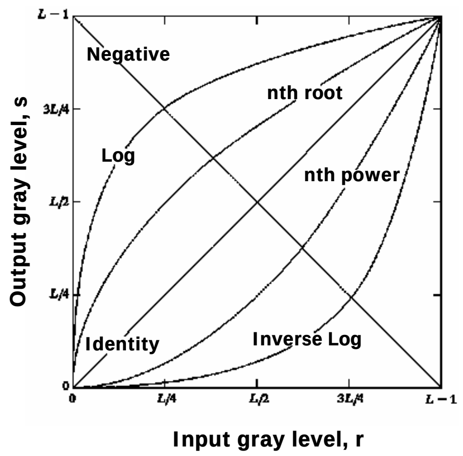
- Neighborhood is bigger than 1x1 pixel
- The value of the mask coefficients determine the nature of the process
- Used in techniques
  - Image Sharpening
  - Image Smoothing

3 basic gray-level transformation functions



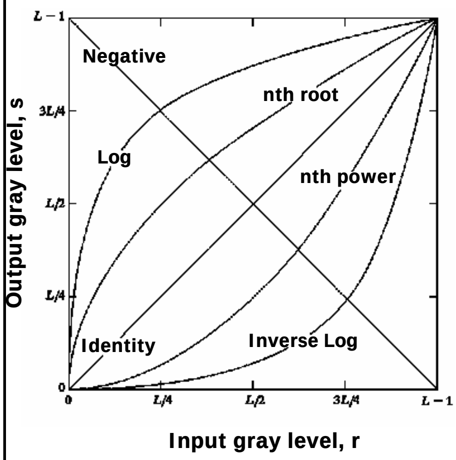
- Linear function
  - Negative and identity transformations
- Logarithm function
  - Log and inverse-log transformation
- Power-law function
  - n^{th} power and n^{th} root transformations

Identity function



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

Image Negatives

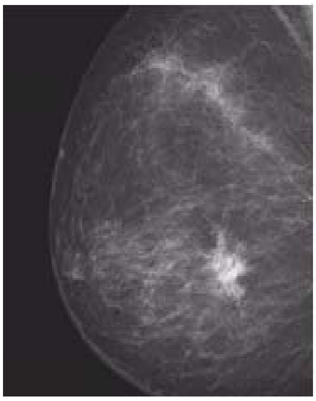


- An image with gray level in the range **[0, L-1]** where $L = 2^n$; n = 1, 2…
- Negative transformation :

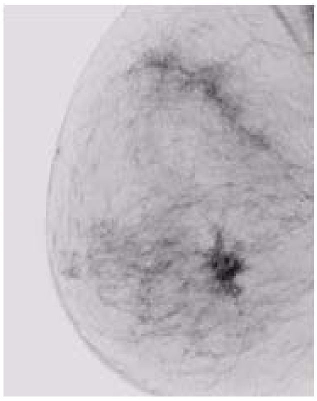
$$s = L - 1 - r$$

- Reversing the intensity levels of an image.
- Suitable for enhancing white or gray detail in dark background.

Example of Negative Image

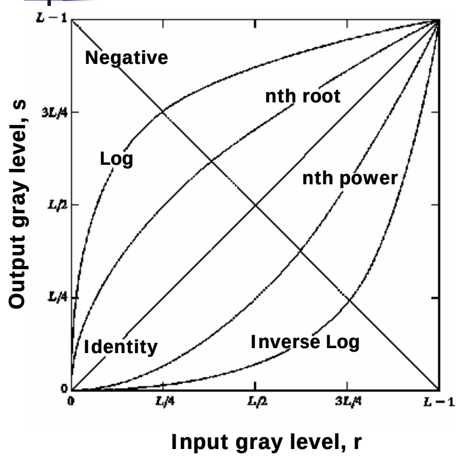


Original Image showing a small lesion



Negative Image : gives a better vision to analyze the image

Log Transformations



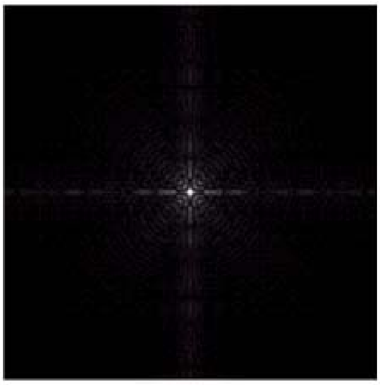
$s = c \log(1+r)$

- 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.
- Expand dark value to enhance details of dark area.

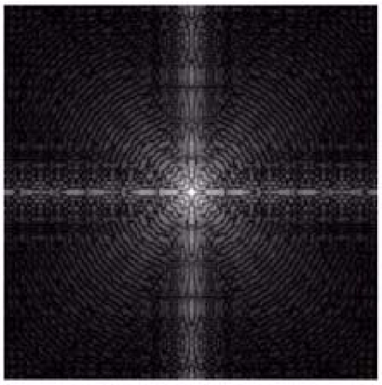
Log Transformations

- 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

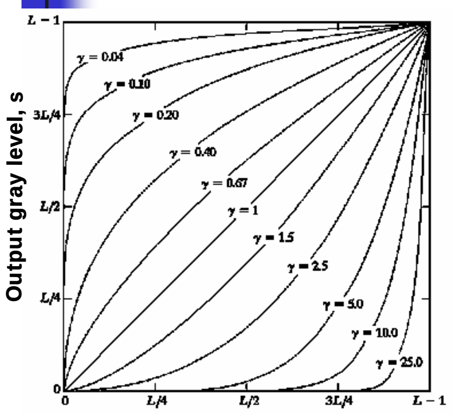


Fourier Spectrum with range = 0 to 1.5 x 10^6



Result after apply the log transformation

Power-Law Transformations

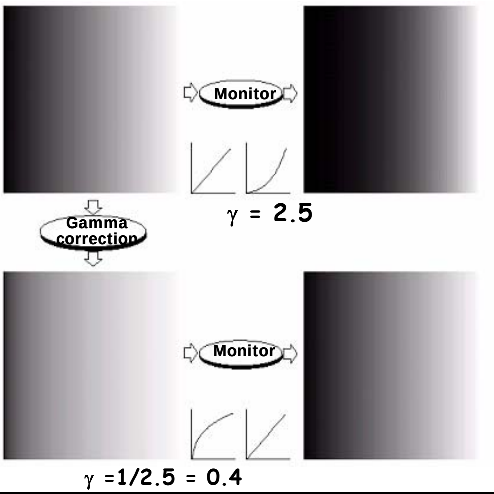


Plots of $s = cr^{\gamma}$ for various values of γ (c = 1 in all cases)

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

- 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

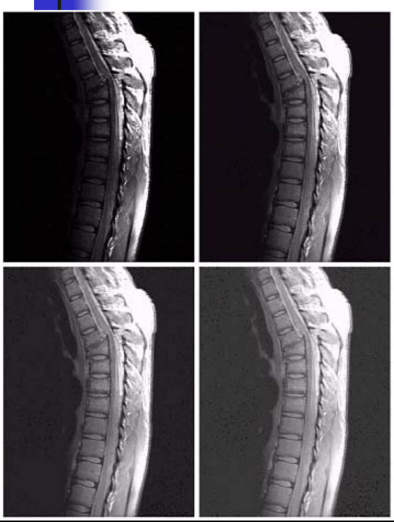
Gamma correction



- 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

- The picture is dark

(b) result after power-law transformation with γ = 0.6, c=1

(c) transformation with γ = 0.4 (best result)

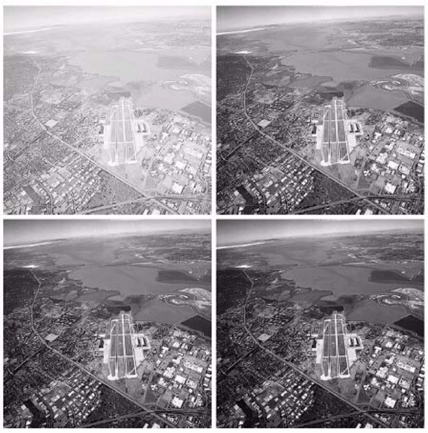
(d) transformation with γ = 0.3 (under acceptable level)

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 $\gamma = 3.0$ (suitable)

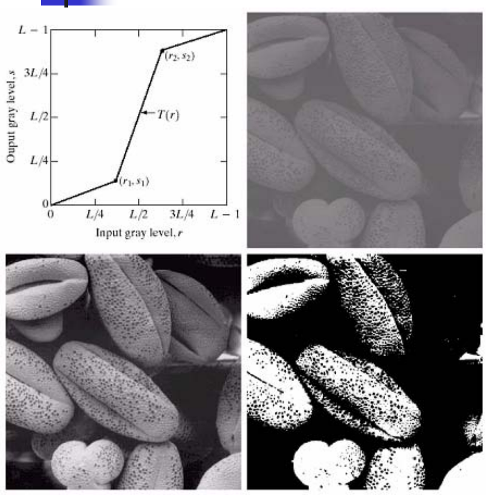
(c) transformation with $\gamma = 4.0$ (suitable)

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

Piecewise-Linear Transformation Functions

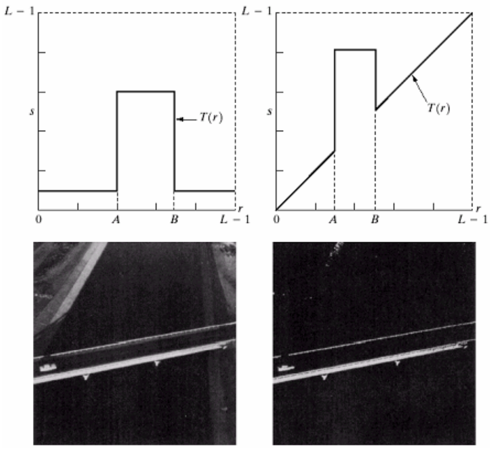
- Advantage:
  - Allow more control on the complexity of T(r).
- Disadvantage:
  - Their specification requires considerably more user input
- Contrast stretching.
- Gray-level slicing.
- Bit-plane slicing.

Contrast Stretching



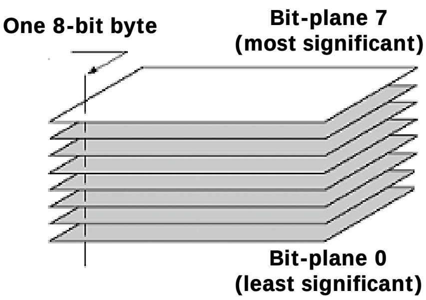
- Increase the dynamic range of gray levels.
- (a) Transformation Function
- (b) a low-contrast image : result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture of image acquisition
- (c) result of contrast stretching
- (d) result of thresholding

Gray-level slicing



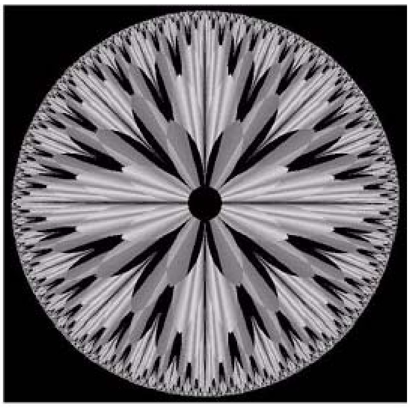
- Highlighting a specific range of gray levels
  - Display a high value of all gray levels in the region of interest and a low value for all other gray levels
- (a) transformation highlights range [A,B] of gray level and reduces all others to a constant level
- (b) transformation highlights range [A,B]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

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

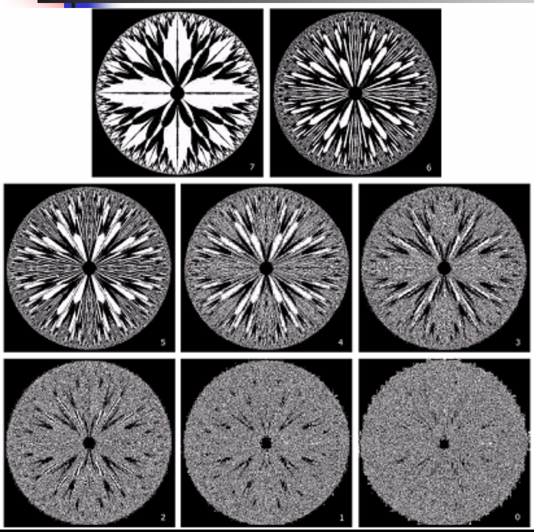
Example



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



Bit-plane 7	Bit-plane 6	
Bit-plane 5	Bit-plane 4	Bit-plane 3
Bit-plane 2	Bit-plane 1	Bit-plane 0

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 k^{th} 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 value at gray level r_k by the total number of pixels in the image, n

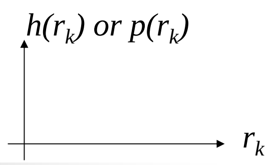
$$p(r_k) = n_k / n$$

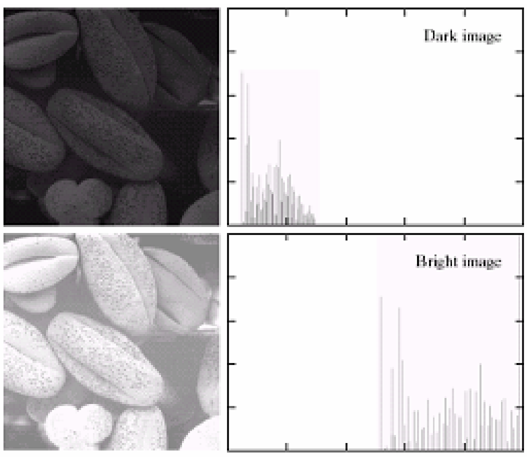
- For k = 0,1,…,L-1
- $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k
- The sum of all components of a normalized histogram is equal to 1

Histogram Processing

- Used effectively for image enhancement
- Information inherent in histograms also is useful in image compression and segmentation
- Data-dependent pixel-based image enhancement method.

Example





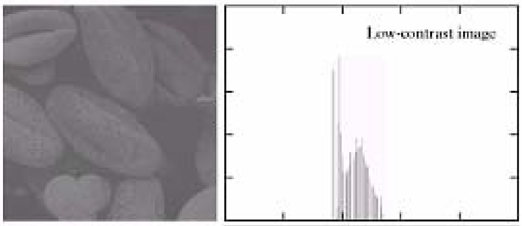
Dark image

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

Bright image

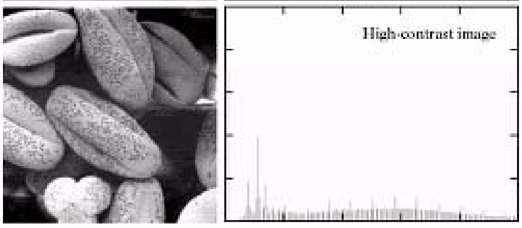
Components of histogram are concentrated on the high side of the gray scale.

Example



Low-contrast image

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

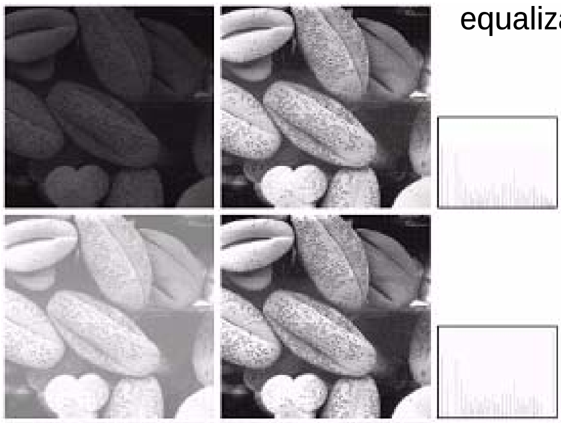


High-contrast image

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

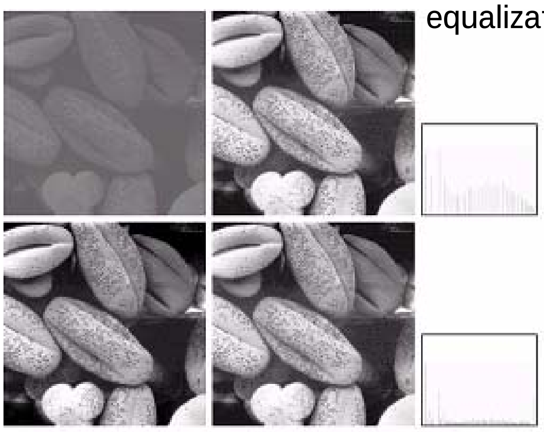
Example

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

Histogram Equalization: Implementation

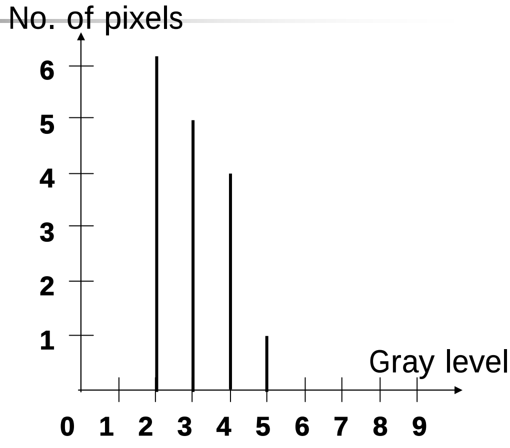
1. Obtain the histogram of the input image.
2. For each input gray level *k*, compute the cumulative sum.
3. For each gray level *k*, scale the sum by (max gray level)/(number of pixels).
4. Discretize the result obtained in 3.
5. Replace each gray level *k* in the input image by the corresponding level obtained in 4.

Example

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

4x4 image

Gray scale = [0,9]



histogram

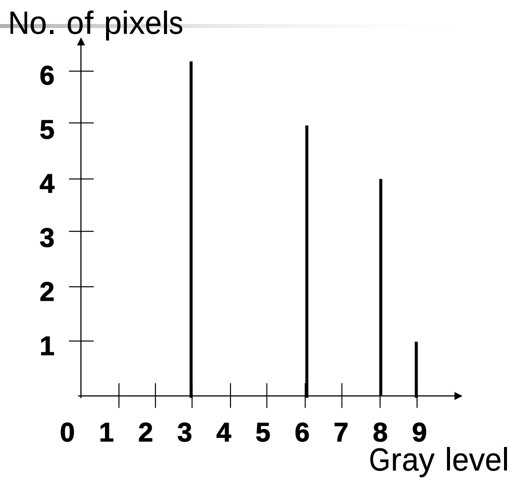
Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
$s \times 9$	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

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
  - 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 Equalization

- A gray-level transformation method that forces the transformed gray level to spread over the entire intensity range.
  - Fully automatic,
  - Data dependent,
  - Contrast enhanced.
- Usually, the discrete-valued histogram equalization algorithm does not yield exact uniform distribution of histogram.
- In practice, one may prefer "histogram specification".

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

Procedure Conclusion

Indirect Method:

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

$$s = T(r)$$

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

$$v = G(z)$$

Procedure Conclusion

3. Set *v = s* to 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

Histogram Matching: Example

- Consider an 8-level image with the shown histogram

0	760
1	1023
2	870
3	660
4	331
5	249
6	122
7	81

- Match it to the image with the histogram

0	0
1	0
2	0
3	615
4	819
5	1229
6	819
7	614

Histogram Matching: Example

1. Equalize the histogram of the input image using transform $s = T(r)$.

k	r_k	n_k	$p(r_k)=n_k/n$	s_k	$7*s_k$	Gray Level	$s_k = T(r_k)$
0	0/7	760	0.185546875	0.185546875	1.298828125	1	1/7
1	1/7	1023	0.249755859	0.435302734	3.047119141	3	3/7
2	2/7	870	0.212402344	0.647705078	4.533935547	5	5/7
3	3/7	660	0.161132813	0.808837891	5.661865234	6	6/7
4	4/7	331	0.080810547	0.889648438	6.227539063	6	6/7
5	5/7	249	0.060791016	0.950439453	6.653076172	7	7/7
6	6/7	122	0.029785156	0.980224609	6.861572266	7	7/7
7	7/7	81	0.019775391	1	7	7	7/7

Histogram Matching: Example

2. Equalize the desired histogram $v = G(z)$.

k	z_k	n_k	$p(z_k)$	v_k	$7*v_k$	Gray Level	$v_k = G(z_k)$
0	0/7	0	0.00	0.00	0	0	0/7
1	1/7	0	0.00	0.00	0	0	0/7
2	2/7	0	0.00	0.00	0	0	0/7
3	3/7	615	0.15	0.15	1.051025391	1	1/7
4	4/7	819	0.20	0.35	2.450683594	2	2/7
5	5/7	1229	0.30	0.65	4.551025391	5	5/7
6	6/7	819	0.20	0.85	5.951025391	6	6/7
7	7/7	614	0.15	1.00	7.001025391	7	7/7

Histogram Matching: Example

3. Set $v = s$ to obtain the composite transform $z = G^{-1}(s) = G^{-1}[T(r)]$

k	r_k	$s_k = T(r_k)$	$z_k = G^{-1}(T(r_k))$	Gray Level
0	0/7	1/7	3/7	3
1	1/7	3/7	4/7	4
2	2/7	5/7	5/7	5
3	3/7	6/7	6/7	6
4	4/7	6/7	6/7	6
5	5/7	7/7	7/7	7
6	6/7	7/7	7/7	7
7	7/7	7/7	7/7	7

Example

Image of Mars moon

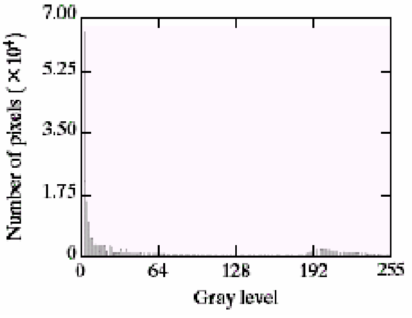
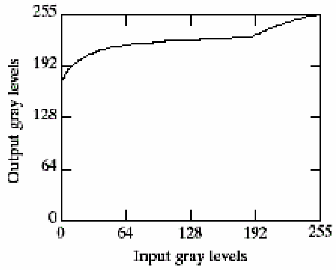
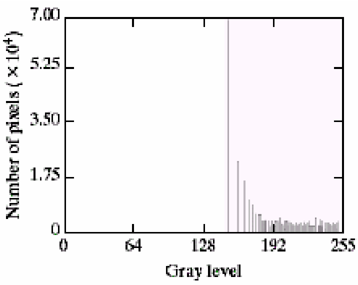


Image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in pixels in the dark end of the gray scale

Image Equalization



Transformation function for histogram equalization



Histogram of the result image

Result image after histogram equalization

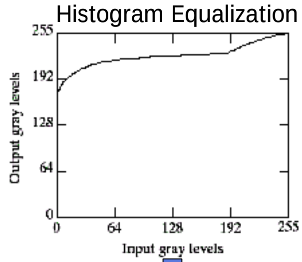
The histogram equalization doesn't make the result image look better. As a consequence, the output image is light and has a washed-out appearance.

Solve the problem

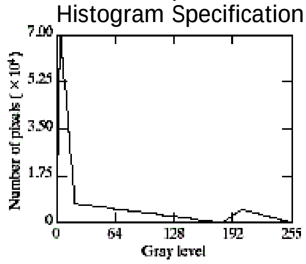
- Since the problem with the transformation function of the histogram equalization was caused by a large concentration of pixels in the original image with levels near 0

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

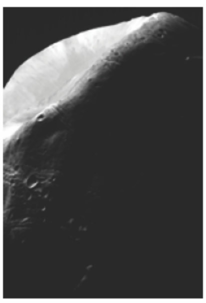
Histogram Equalization



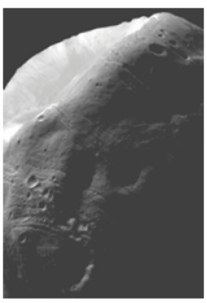
Histogram Specification



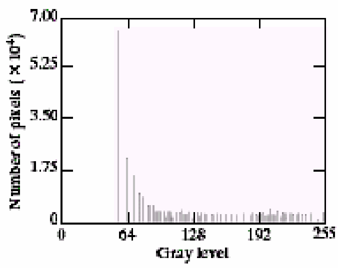
Result image and its histogram



Original image



After histogram Specification



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.

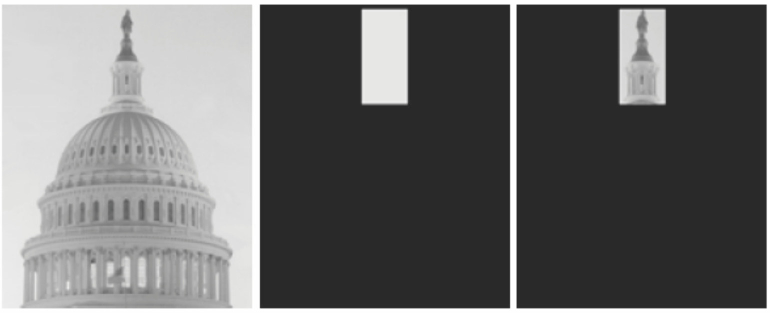
Enhancement using Arithmetic/Logic Operations

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

Logic Operations

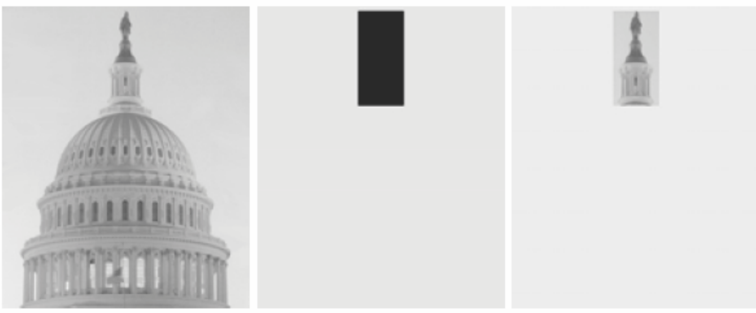
- Logic operation is performed on gray-level images, the pixel values are processed as binary numbers
- NOT operation = negative transformation

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

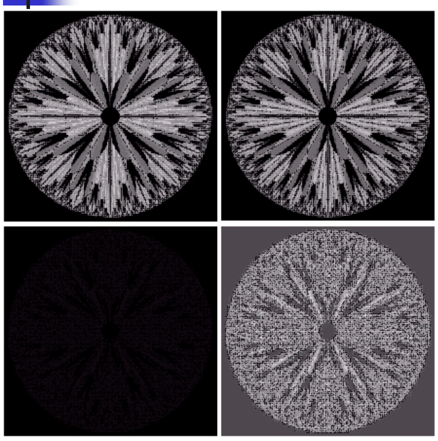
Image Subtraction

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

- 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 details
  - 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)

Note

- We may have to adjust the gray-scale of the subtracted image to be [0, 255] (if 8-bit is used)
- 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 modeled as:

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

Where $f(x,y)$ is the original image, and $\eta(x,y)$ is an uncorrelated zero-mean noise process

- Objective: to reduce the noise content by averaging a set of noisy images

Image Averaging

- Define an image formed by averaging K different noisy images:

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

- It follows that:

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

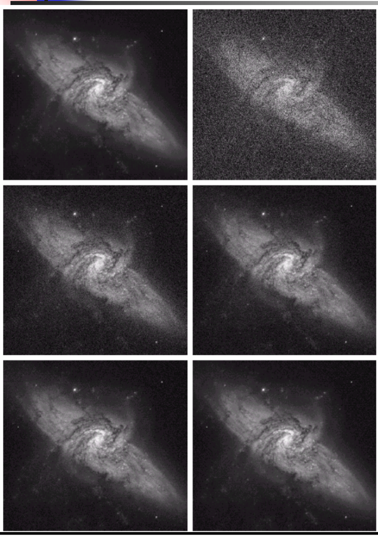
= expected value of g (output after averaging) = original image f(x,y)

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.

Example

a	b
c	d
e	f



- 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

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Spatial Filtering

- 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.

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn}$$
$$= \sum_{i=1}^{mn} w_i z_i$$

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
- reducing the rapid pixel-to-pixel variation (high frequency) in gray values.

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.
- sharp details are lost.

Smoothing Linear Filters

- 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 (may be undesirable)

3x3 Smoothing Linear Filters

$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

box filter

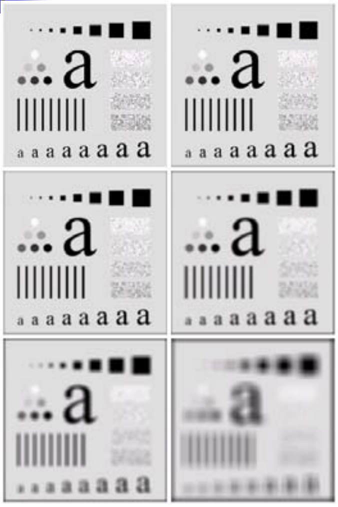
weighted average

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

(reduce blurring in the smoothing process)

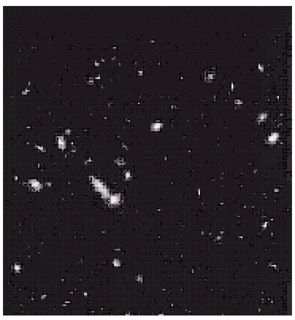
Example

a	b
c	d
e	f

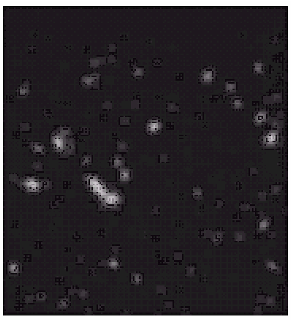


- a). original image 500x500 pixel
- b). - f). results of smoothing with square averaging filter masks of size n = 3, 5, 9, 15 and 35, respectively.
- Note:
  - big mask is used to eliminate small objects from an image.

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)

- Nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the filter mask and then replacing the value of the center pixel with the result of the ranking operation

- example
  - median filter : R = median{z_k |k = 1,2,…,n x n}
  - max filter : R = max{z_k |k = 1,2,…,n x n}
  - min filter : R = min{z_k |k = 1,2,…,n x n}
- note: n x n is the size of the mask

Median Filters

- popular for certain types of random noise
  - impulse noise ⇨ salt and pepper noise

- they provide excellent noise-reduction capabilities, with considering less blurring than linear filters of similat size.

- forces the points with distinct gray levels to be more like their neighbors.

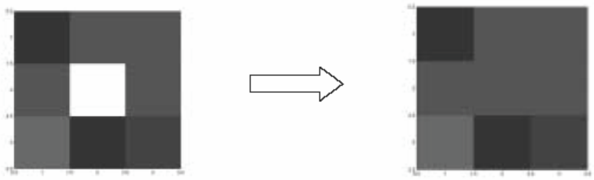
Median Filtering: Example

10	20	20
20	100	20
25	10	15

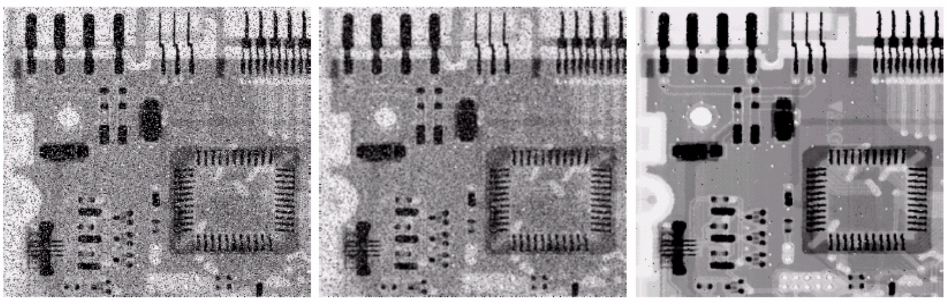
[10,10,15,20,20,20,20,25,100]

median value

Therefore, replace 100 with 20



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 an effect of a 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 similar to integration
- thus, we can guess that the sharpening must be accomplished by spatial differentiation.

First-order derivative (1D)

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

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

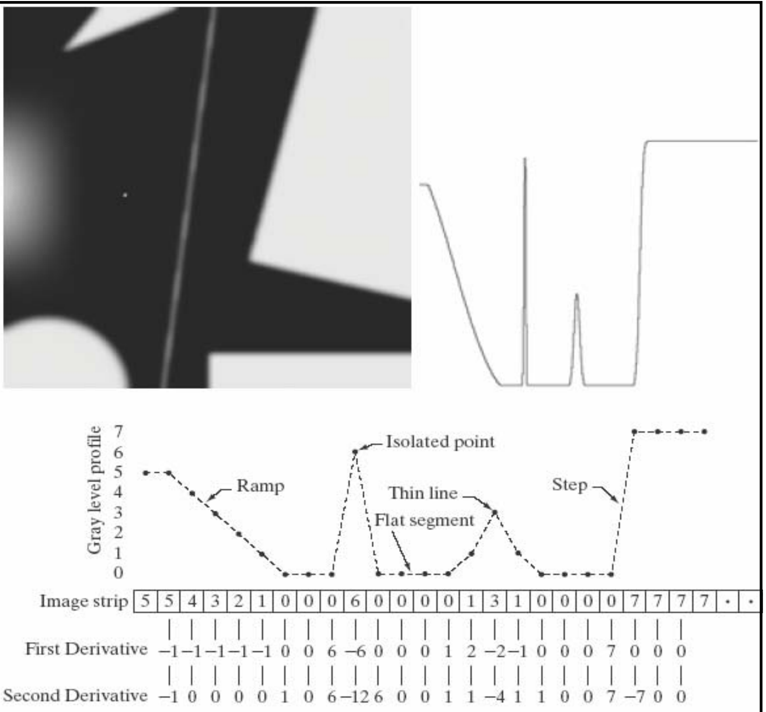
Second-order derivative (1D)

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

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

a b
c

FIGURE 3.38 (a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).



First and Second-order derivative of f(x,y) (2D)

- 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.

Laplacian Operator

- Isotropic filters: response is independent of direction (rotation-invariant).

- The simplest isotropic derivative operator is the Laplacian

To get a sharp image:

- 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

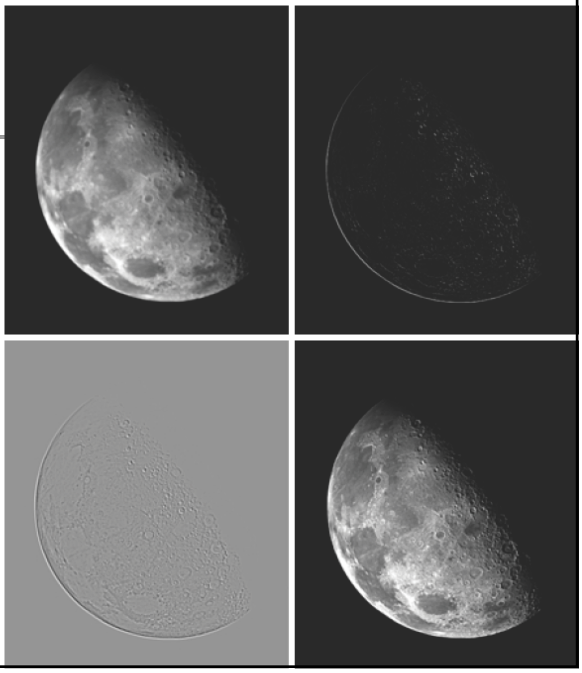
if the center coefficient of the Laplacian mask is positive

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 subtraction 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

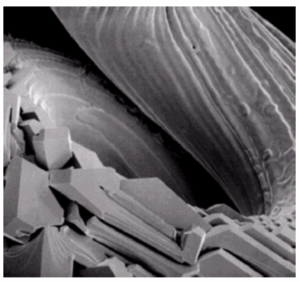
$$
\begin{aligned}
g(x,y) = f(x,y) - [&f(x+1,y) + f(x-1,y) \\
&+ f(x,y+1) + f(x,y-1) - 4f(x,y)] \\
= 5f(x,y) - [&f(x+1,y) + f(x-1,y) \\
&+ f(x,y+1) + f(x,y-1)]
\end{aligned}
$$

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



a b c
d e

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

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

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

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

Unsharp masking

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

sharpened image = original image – blurred image

- An image can be sharpened by subtracting a blurred version of it from the original image

High-boost filtering

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y) \qquad (A \geq 1)$$
$$= (A-1)f(x,y) + f(x,y) - \bar{f}(x,y)$$
$$= (A-1)f(x,y) + f_s(x,y)$$

- 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

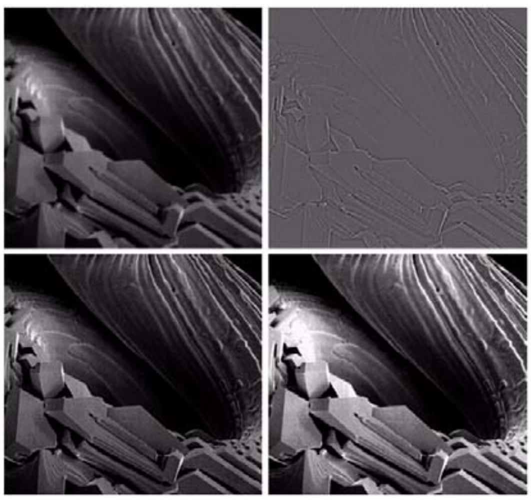
−1	−1	−1
−1	$A+8$	−1
−1	−1	−1

- $A \geq 1$
- 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 $A = 1.7$.



Use of First Derivatives for Enhancement-The Gradient

- First derivatives in image processing are implemented using the magnitude of the gradient.

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

Gradient Operator

- Magnitude of the gradient.

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

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

the magnitude becomes nonlinear

commonly approx.

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

- Simpler to compute
- Still preserves relative changes in gray levels

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- simplest approximation, 2x2

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

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

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

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- 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
- An approximation using absolute values

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$
$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

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

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

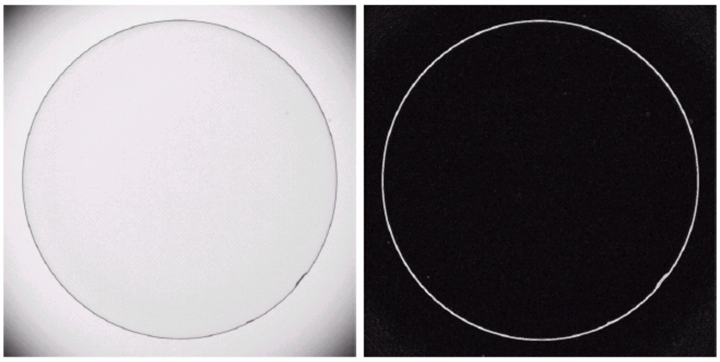
-1	-2	-1
0	0	0
1	2	1

-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.

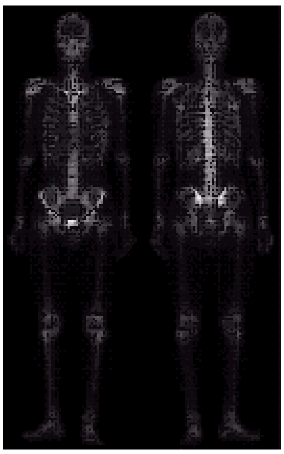
Example



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.)

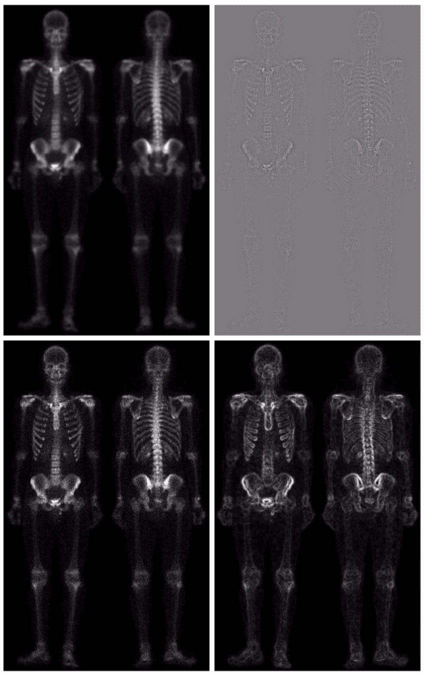
Example of Combining Spatial Enhancement Methods



- 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

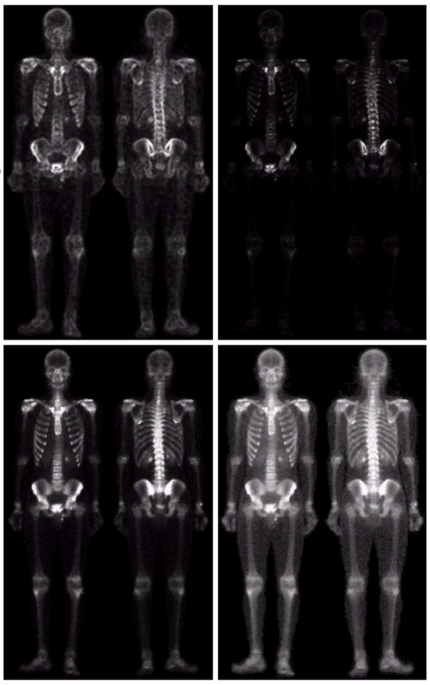
Example of Combining Spatial Enhancement Methods

- 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).



e f
g h

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.)