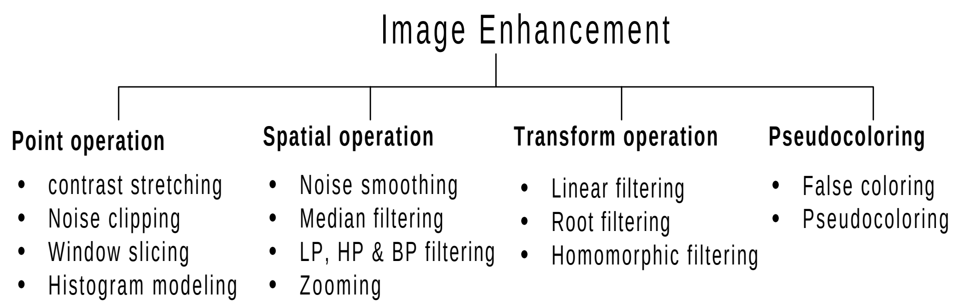
Image Enhancement

- Introduction
- Enhancement by point processing
  - Simple intensity transformation
  - Histogram processing
- Spatial filtering
  - Smoothing filters
  - Sharpening filters
- Enhancement in the frequency domain
- Pseudo-color image processing

1. Introduction

- The principal objective of image enhancement is to process a given image so that the result is more suitable than the original image for a specific application.
- It accentuates or sharpens image features such as edges, boundaries, or contrast to make a graphic display more helpful for display and analysis.
- The enhancement doesn't increase the inherent information content of the data, but it increases the dynamic range of the chosen features so that they can be detected easily.



- The greatest difficulty in image enhancement is quantifying the criterion for enhancement and, therefore, a large number of image enhancement techniques are empirical and require interactive procedures to obtain satisfactory results.
- Image enhancement methods can be based on either spatial or frequency domain techniques.

Spatial domain enhancement methods:

- Spatial domain techniques are performed to the image plane itself and they are based on direct manipulation of pixels in an image.
- The operation can be formulated as $g(x,y) = \mathbf{T}[f(x,y)]$, where g is the output, f is the input image and $\mathbf{T}$ is an operation on f defined over some neighborhood of (x,y).
- According to the operations on the image pixels, it can be further divided into 2 categories: *Point operations* and *spatial operations* (including linear and non-linear operations).

Frequency domain enhancement methods:

- These methods enhance an image $f(x,y)$ by convoluting the image with a linear, position invariant operator.
- The 2D convolution is performed in frequency domain with DFT.

Spatial domain: $g(x,y)=f(x,y)*h(x,y)$

Frequency domain: $G(w_1,w_2)=F(w_1,w_2)H(w_1,w_2)$

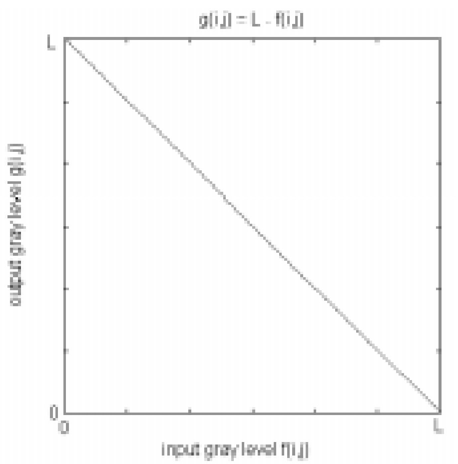
2. Enhancement by point processing

- These processing methods are based only on the intensity of single pixels.

2.1 Simple intensity transformation:

(a). *Image negatives*:

- Negatives of digital images are useful in numerous applications, such as displaying medical images and photographing a screen with monochrome positive film with the idea of using the resulting negatives as normal slides.
- Transform function T : g(x,y)=L-f(x,y), where L is the max. intensity.

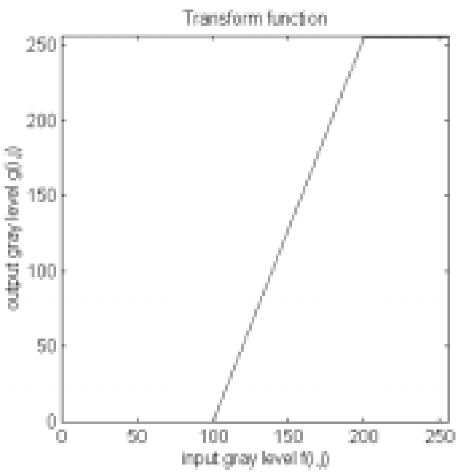


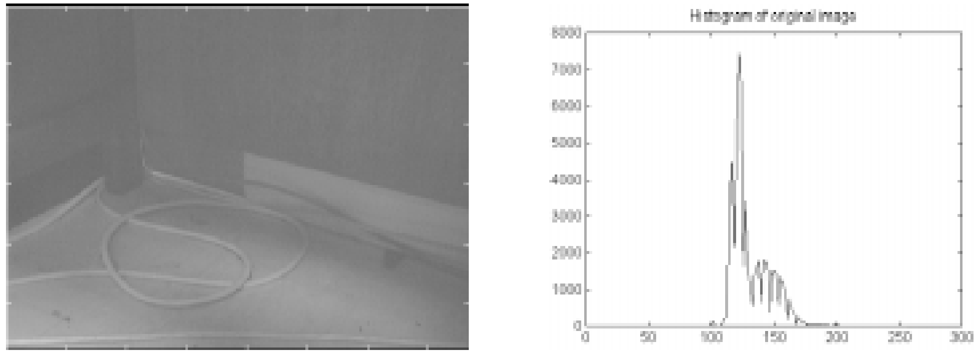
Original

Negative

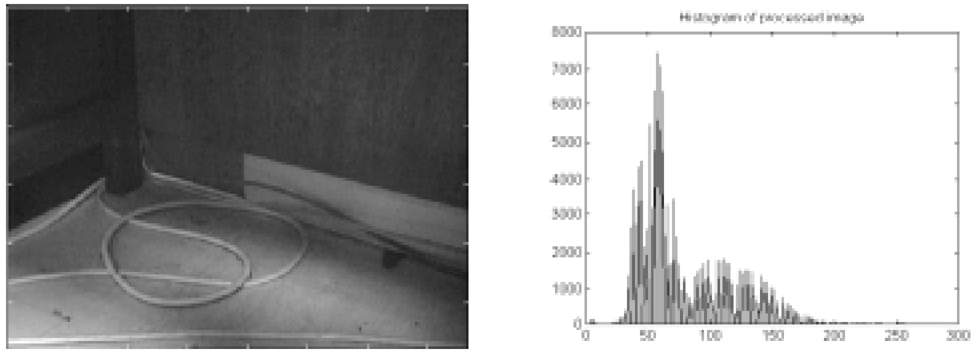
(b). *Contrast stretching*

- Low-contrast images can result from poor illumination, lack of dynamic range in the image sensor, or even wrong setting of a lens aperture during image acquisition.
- The idea behind contrast stretching is to increase the dynamic range of the gray levels in the image being processed.





Original



Processed image

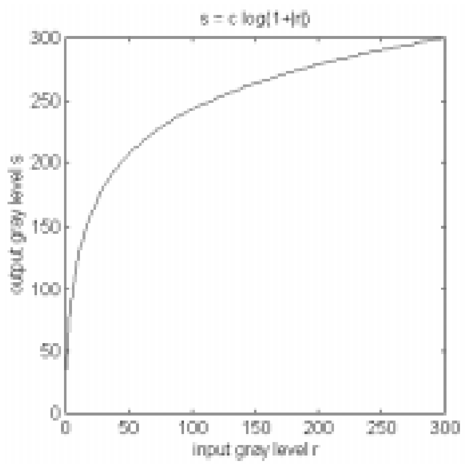
- Special case: If $r_1=r_2=0$, $s_1=0$ and $s_2=L-1$, then it is actually a thresholding that creates a binary images.

(c). *Compression of dynamic range*

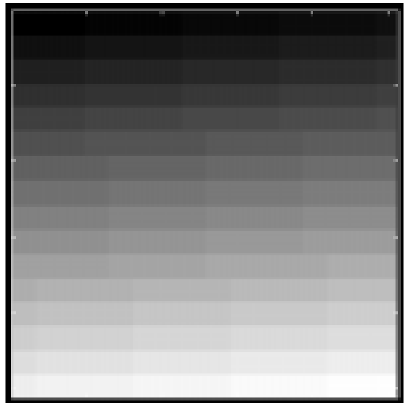
- Sometimes the dynamic range of a processed image far exceeds the capability of the display device, in which case only the brightest parts of the images are visible on the display screen.
- An effective way to compress the dynamic range of pixel values is to perform the following intensity transformation function:

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

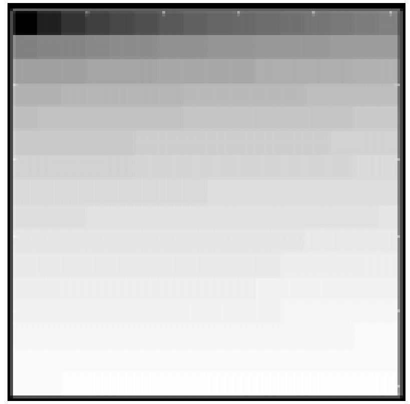
where c is a scaling constant, and the logarithm function performs the desired compression.



Transform function



Original



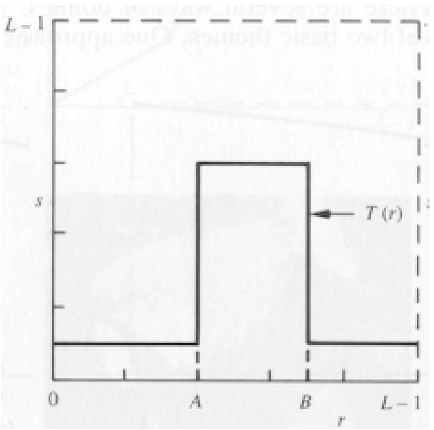
Processed output

(d) *Gray-level slicing*

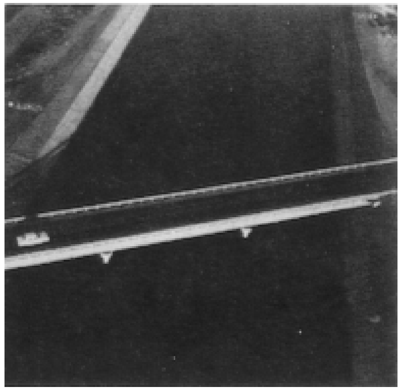
- Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in x-ray images.

• Example 1:

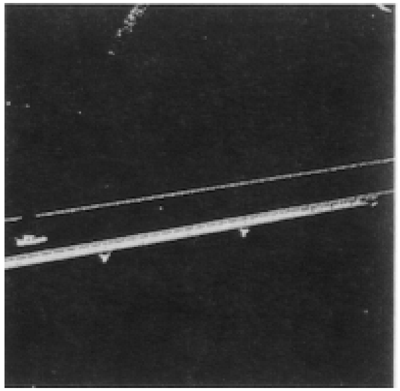
A transformation function that highlights a range [A,B] of intensities while diminishing all others to a constant.



(a)



(b)

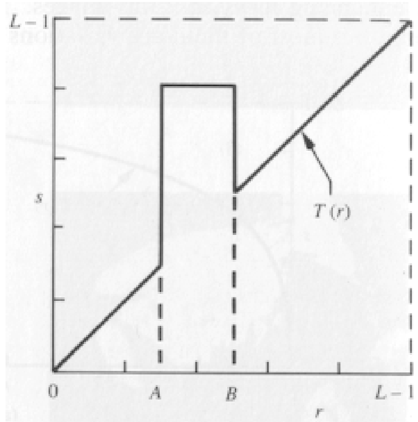


(c)

Fig 1. (a) Transfer function, (b) Original image, (c) Processing output.

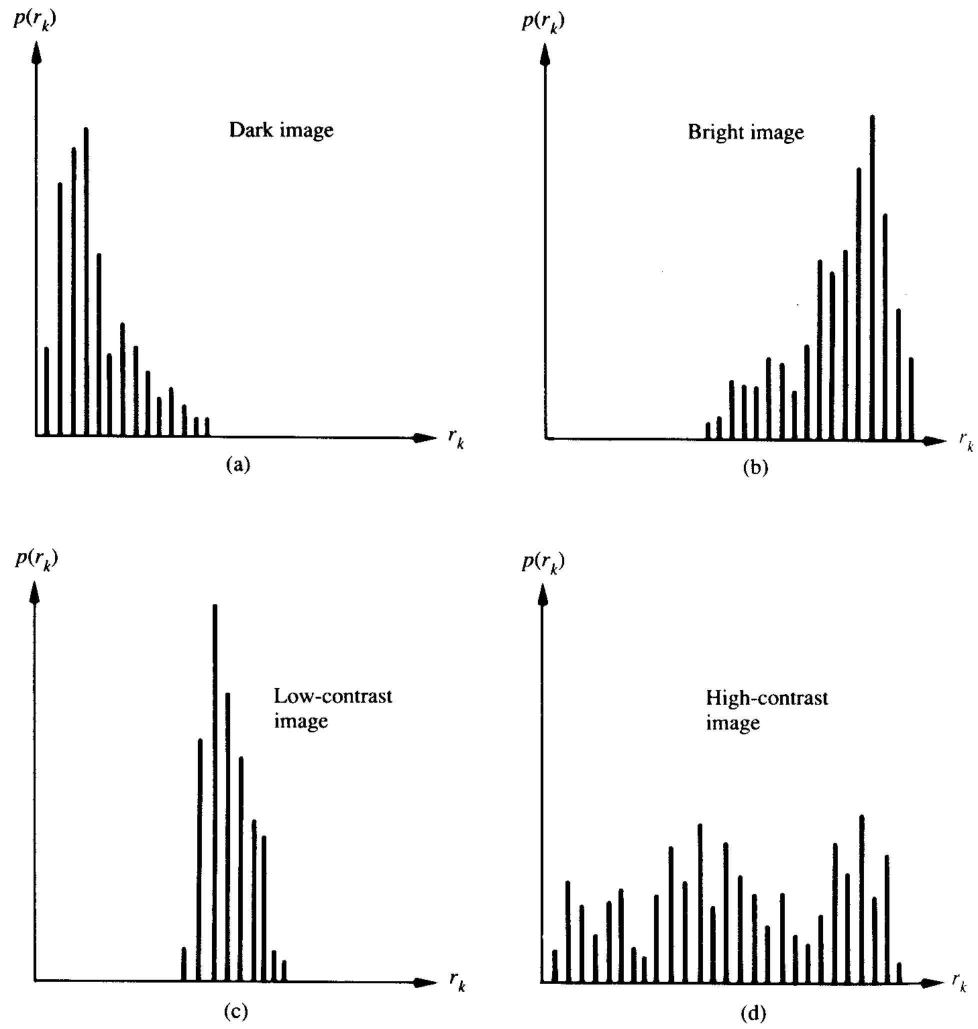
• Example 2:

A transformation function that highlights a range [A,B] of intensities but preserves all others.



2.2 Histogram processing:

- The histogram of a digital image with gray levels in the range [0,L-1] is a discrete function $p(r_k)=n_k/n$, where r_k is the kth gray level, n_k is the number of pixels in the image with that gray level, n is the total number of pixels in the image, and k=0,1..L-1.
- $P(r_k)$ gives an estimate of the probability of occurrence of gray level r_k.
- The shape of the histogram of an image gives us useful information about the possibility for contrast enhancement.
- A histogram of a narrow shape indicates little dynamic range and thus corresponds to an image having low contrast.



(a) *Histogram equalization*

- The objective is to map an input image to an output image such that its histogram. is uniform after the mapping.
- Let r represent the gray levels in the image to be enhanced and s is the enhanced output with a transformation of the form $s=\mathbf{T}(r)$.
- Assumption:
  1. $\mathbf{T}(r)$ is single-valued and monotonically increasing in the interval [0,1], which preserves the order from black to white in the gray scale.
  2. $0 \le T(r) \le 1$ for $0 \le r \le 1$, which guarantees the mapping is consistent with the allowed range of pixel values.
- If $P_r(r)$ and $\mathbf{T}(r)$ are known and $\mathbf{T}^{-1}(s)$ satisfies condition (a), the pdf of the transformed gray levels is $P_s(s) = P_r(r)\left.\frac{dr}{ds}\right|_{r=T^{-1}(s)}$
- If $s = T(r) = \int_0^r P_r(w)dw$ for $0 \le r \le 1$, then we have $\frac{ds}{dr} = P_r(r)$ and hence $P_s(s) = 1$ for $0 \le s \le 1$.

- Using a transformation function equal to the cumulative distribution of r produces an image whose gray levels have a uniform density, which implies an increase in the dynamic range of the pixels.
- In order to be useful for digital image processing, eqns. should be formulated in discrete form:

$$P_r(r_k) = \frac{n_k}{n} \text{ and } s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n}, \text{ where k=0,1...L-1}$$

- A plot of $P_r(r_k)$ versus r_k is actually a histogram, and the technique used for obtaining a uniform histogram is known as histogram equalization or histogram linearization.

- Example: Equalizing an image of 6 gray levels.

Index k	0	1	2	3	4	5
Normalized Input level, $r_k/5$	0.0	0.2	0.4	0.6	0.8	1.0
Freq. Count of r_k, n_k	**4**	**7**	**2**	**1**	**0**	**1**
Probability $P(r_k) = n_k/n$	4/15	7/15	2/15	1/15	0/15	1/15
$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n}$	4/15 = **0.27**	11/15 = **0.73**	13/15 = **0.87**	14/15 = **0.93**	14/15 = **0.93**	15/15 = **1.00**
Quantized s_k	0.2	0.8	0.8	1.0	1.0	1.0

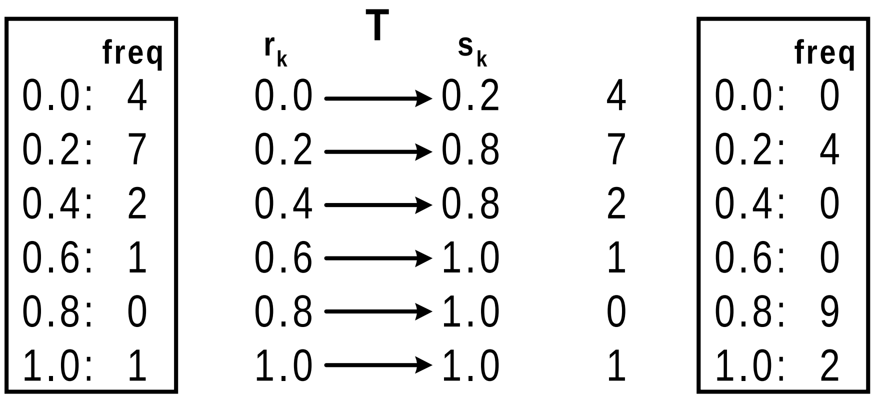
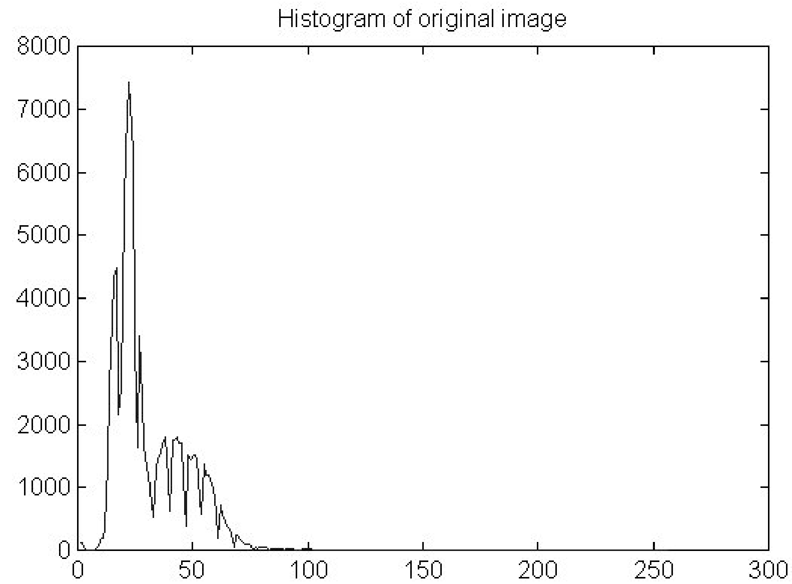
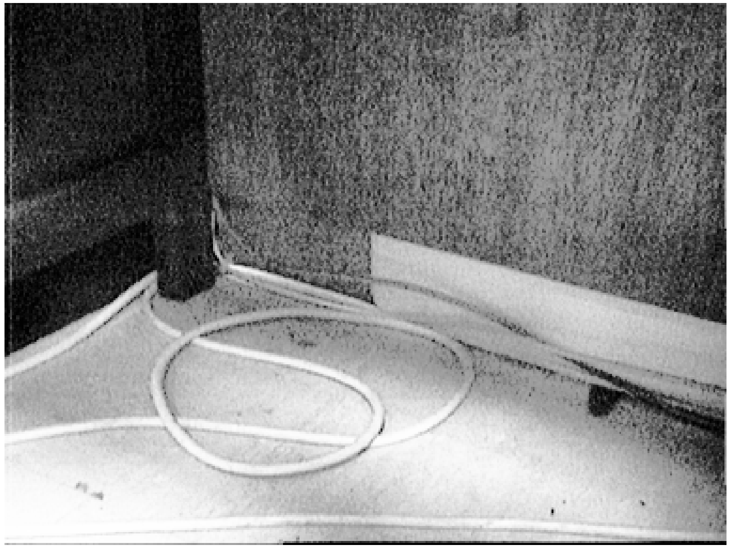


Fig 2. Example of histogram equalization

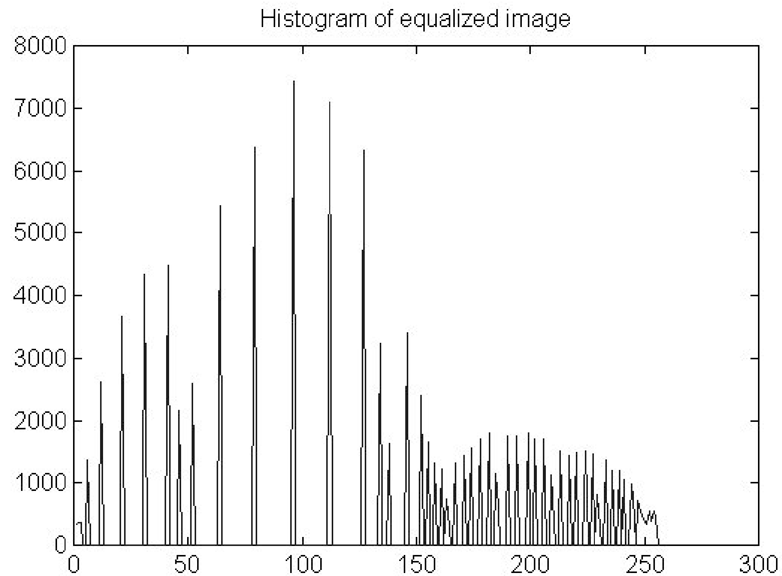
Original image



Histogram



Equalized image



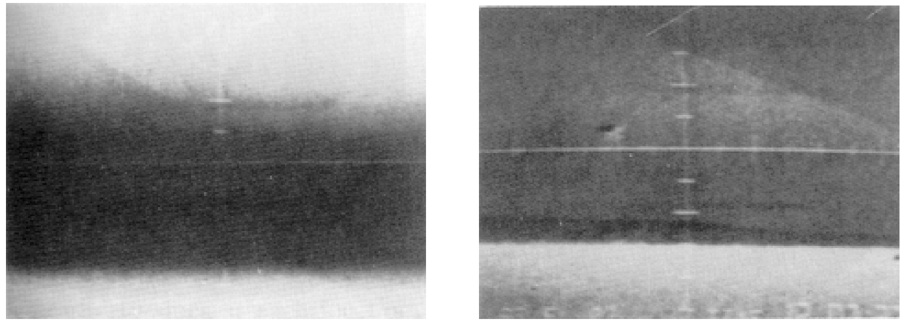
Histogram

(b) ***Histogram specification***

- Histogram equalization only generates an approximation to a uniform histogram.
- Sometimes the ability to specify particular histogram shapes capable of highlighting certain gray-level ranges in an image is desirable.
- Procedures:
  1. Determine the transformation $s_k = T(r_k)$ that can equalize the original image's histogram $p_r(r)$.
  2. Determine the transformation $s_k = G(b_k)$ that can equalize the desired image's histogram $p_b(b)$.
  3. Perform transformation $G^{-1}(T(r_k))$.
- The principal difficulty in applying the histogram specification method to image enhancement lies in being able to construct a meaningful histogram.

(c) *Local enhancement*

- It is often necessary to enhance details over small areas.
- The number of pixels in these areas may have negligible influence on the computation of a global transformation, so the use of global histogram specification does not necessarily guarantee the desired local enhancement.
- Procedures:
  1. Define a square or rectangular neighborhood and move the center of this window from pixel to pixel.
  2. Determine the histogram equalization or histogram specification transformation function with the histogram of the windowed image at each location.
  3. Map the gray level centers in the window with the transformation function.
  4. Move to an adjacent pixel location and the procedure is repeated.



(a) (b)

Fig 3. Image before and after local enhancement.

3. Spatial Filtering:

- The use of spatial masks for image processing is called *spatial filtering*.
- The masks used are called *spatial filters*.

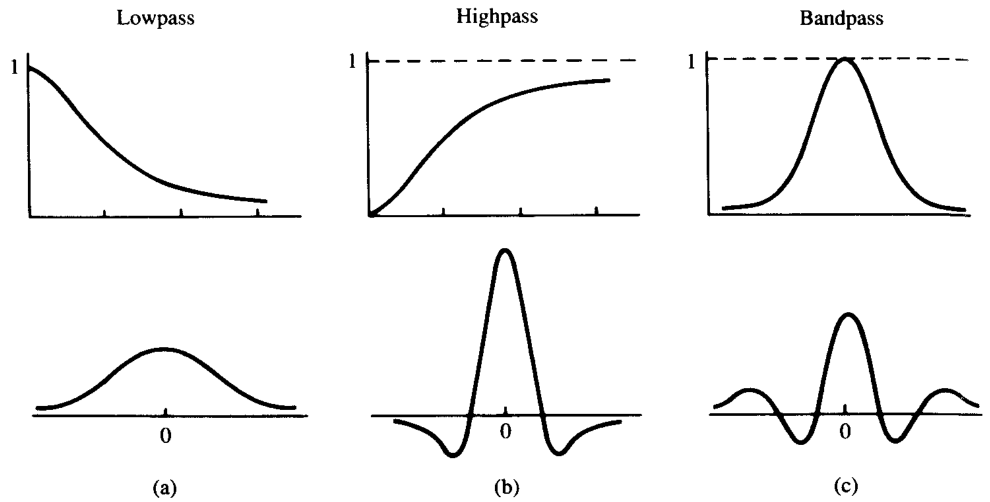


Fig 4. Top: cross sections of basic shapes for circularly symmetric frequency domain filter. Bottom: cross sections of corresponding spatial domain filters. (a) lowpass, (b) bandpass and (c) highpass filters.

- The basic approach is to sum products between the mask coefficients and the intensities of the pixels under the mask at a specific location in the image. (2D convolution)

$$R(x, y) = \sum_{i=-d}^{d} \sum_{-d}^{d} w(i, j) f(x-i, y-j)$$

where (2d+1)X(2d+1) is the mask size, *w(i,j)*'s are weights of the mask, *f(x,y)* is input pixel at coordinates (x,y), *R(x,y)* is the output value at (x,y).

*

- If the center of the mask is at location (x,y) in the image, the gray level of the pixel located at (x,y) is replaced by R, the mask is then moved to the next location in the image and the process is repeated. This continues until all pixel locations have been covered.

3.1. Smoothing filter:

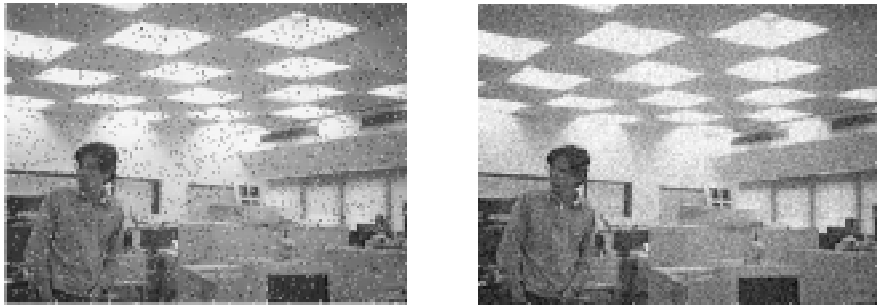
- Smoothing filters are used for blurring and for noise reduction.
- Blurring is used in preprocessing steps, such as removal of small details from an image prior to object extraction, and bridging of small gaps in lines or curves.
- Noise reduction can be accomplishing by blurring with a linear filter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

(a). *Low pass filtering*

- The key requirement is that all coefficients are positive.
- Neighborhood averaging is a special case of LPF where all coefficients are equal.
- It blurs edges and other sharp details in the image.
- Example: $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

(b). *Median filtering*

- If the objective is to achieve noise reduction instead of blurring, this method should be used.
- This method is particularly effective when the noise pattern consists of strong, spike-like components and the characteristic to be preserved is edge sharpness.
- It is a nonlinear operation.
- For each input pixel f(x,y), we sort the values of the pixel and its neighbors to determine their median and assign its value to output pixel g(x,y).



Original with (a) spike noise (b) white noise

Median filtering output

Low-pass filtering output

3.2. Sharpening 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.
- Uses of image sharpening vary and include applications ranging from electronic printing and medical imaging to industrial inspection and autonomous target detection in smart weapons.

(a). *Basic highpass spatial filter*

- The shape of the impulse response needed to implement a highpass spatial filter indicates that the filter should have positive coefficients near its center, and negative coefficients in the outer periphery.
- Example : filter mask of a 3x3 sharpening filter

$$\frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- The filtering output pixels might be of a gray level exceeding [0,L-1].

- The results of highpass filtering involve some form of scaling and/or clipping to make sure that the gray levels of the final results are within [0,L-1].

(b). *Derivative filters*.

- Differentiation can be expected to have the opposite effect of averaging, which tends to blur detail in an image, and thus sharpen an image and be able to detect edges.

- The most common method of differentiation in image processing applications is the gradient.

- For a function *f(x,y)*, the gradient of *f* at coordinates (x',y') is defined as the vector

$$\nabla f(x', y') = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}_{(x', y')}$$

- Its magnitude can be approximated in a number of ways, which result in a number of operators such as Roberts, Prewitt and Sobel operators for computing its value.

Example: masks of various operators

1	0
0	-1

0	1
-1	0

Roberts

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Prewitt

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

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

Sobel

Sobel (V)

Sobel (H)

Sobel

Fig 5. Edge detection with various operators (Outputs are normalized.)

Prewitt

Robert

Fig 5 (Cont.). Edge detection with various operators (Outputs are normalized.)

4. Enhancement in the frequency domain:

- We simply compute the Fourier transform of the image to be enhanced, multiply the result by a filter transfer function, and take the inverse transform to produce the enhanced image.

Spatial domain: $g(x,y)=f(x,y)*h(x,y)$

$$\Updownarrow$$

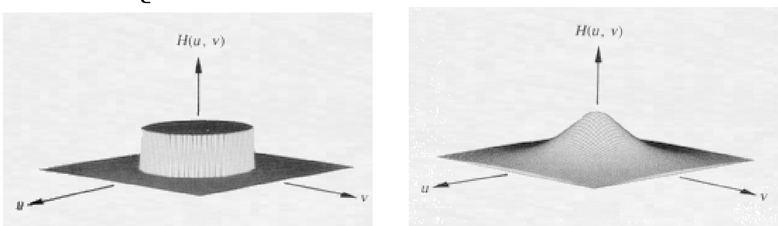
Frequency domain: $G(w_1,w_2)=F(w_1,w_2)H(w_1,w_2)$

Lowpass filtering

- Edges and sharp transitions in the gray levels contribute to the high frequency content of its Fourier transform, so a lowpass filter smoothes an image.

- Formula of ideal LPF

$$H(u,v)=\begin{cases} 1 & if \;\; D(u,v)\le D_o \\ 0 & else \end{cases}$$



(a) (b)

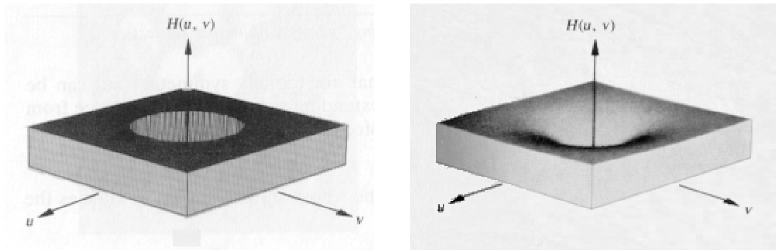
Fig 6. (a) Ideal LPF; (b) Butterworth LPF.

Highpass filtering

- A highpass filter attenuates the low frequency components without disturbing the high frequency information in the Fourier transform domain can sharpen edges.

- Formula of ideal HPF function

$$H(u,v)=\begin{cases} 0 & if \;\; D(u,v)\le D_o \\ 1 & else \end{cases}$$



(a) (b)

Fig 7. (a) Ideal HPF; (b) Butterworth HPF.

5. Pseudo color image processing

- In automated image analysis, color is a powerful descriptor that often simplifies object identification and extraction from a scene.
- Human eye performs much better in discerning shades of color than gray scale.
- A monochrome image can be enhanced by using colors to represent different gray levels or frequencies.

Gray level to color transformation

- To perform 3 independent transformations on the gray level of any input pixel. The three results are then fed separately into the R, G, B guns of a color monitor.

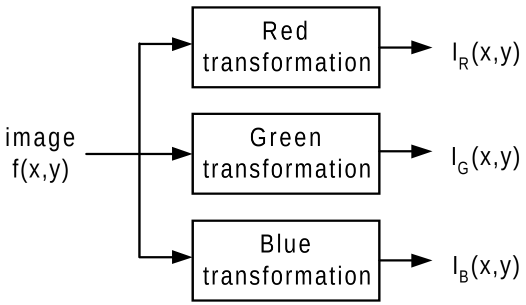


Fig 8. Functional block diagram for pseudo-color image processing.

- This method produces a composite image whose color content is modulated by the nature of the transformation functions.

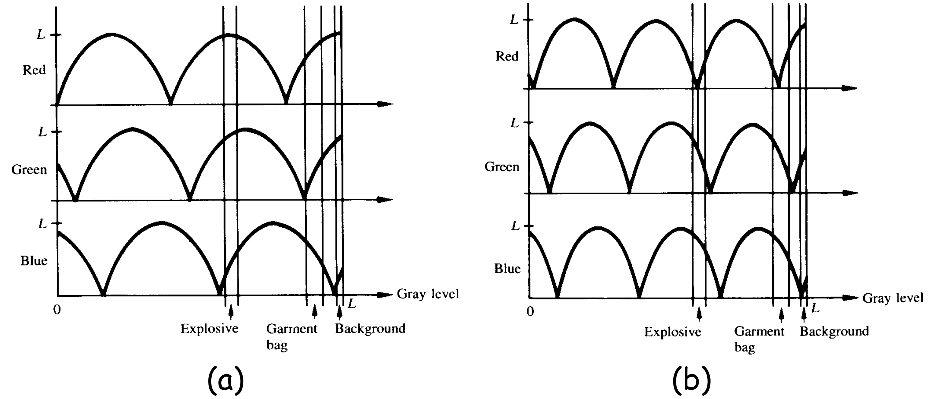
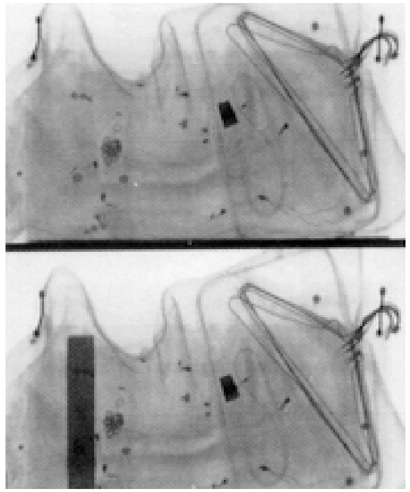


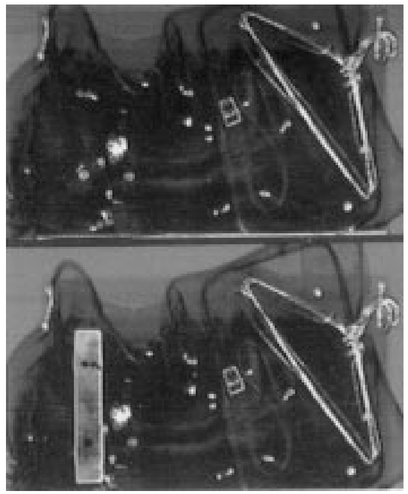
Fig 9. Two examples of transformation functions

- These sinusoidal functions contain regions of relatively constant value around the peaks as well as regions that change rapidly near the valleys.
- Changing the phase and frequency of each sinusoidal can emphasize ranges in the gray scale.
- A small change in the phase between the 3 transformations produces little change in pixels whose gray levels correspond to peaks in the sinusoidals (i.e. Case of $I_R(x,y) \approx I_G(x,y) \approx I_B(x,y)$).

- Pixels with gray level values in the steep section of the sinusoids are assigned a much stronger color content as a result of significant differences between the amplitudes of the 3 sinusoids caused by the phase displacement between them.
  (i.e. Case of $I_R(x,y) \neq I_G(x,y) \neq I_B(x,y)$)



(a)



(b)

(c)

Fig 10. Pseudo-color enhancement by using the gray-level to color transformations. (a) Original image, (b) and (c) results obtained with transformation functions (a) and (b) respectively.