Image transformation and enhancement

Before speak about image transformation. We speak about image noises.

* **Noise is random variation in the brightness occurs due to sensors or circuity of the scanner or electric components**
* type is the noise type

**\_ guassian noise: occurs during** acquisition like  sensor noise or poor illumination and/or high temperature

**\_ posisson noise (shot noise):** variation in the number of photons sensed at a given exposure level

**\_ uniform noise:** The noise caused by [quantizing](https://en.wikipedia.org/wiki/Quantization_(signal_processing)) the pixels of a sensed image to a number of discrete levels is known as [quantization](https://en.wikipedia.org/wiki/Quantization_(image_processing)) noise.

**\_ impulse noise (salt and pepper):** have dark pixels in bright regions and bright pixels in dark regions because ADC errors

• g = imnoise (f, 'gaussian', m, var)

* **g = imnoise (f, 'salt & pepper', d) % d is noise density**
* g = imnoise (f, 'poisson ')

**we focus attention on two important categories of spatial domain processing: intensity (gray-level) transformations and spatial filtering**

* **transformation and filterization are related to image enhancement.**

**spatial domain techniques operate directly on the pixels of an image.**

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

Because the output value depends only on the intensity value at a point, and not on a neighborhood of points, intensity transformation functions frequently are written in simplified form as

s = T(r)

where r denotes the intensity of f and s the intensity of g, both at the same coordinates (x, y) in the images

Functions imadjust and stretchlim

* Imadjust : is the basic Image Processing Toolbox function for intensity transformations of gray-scale images. It has the general syntax

g = imadjust(f, [low\_in high\_in], [low\_out high\_out], gamma)

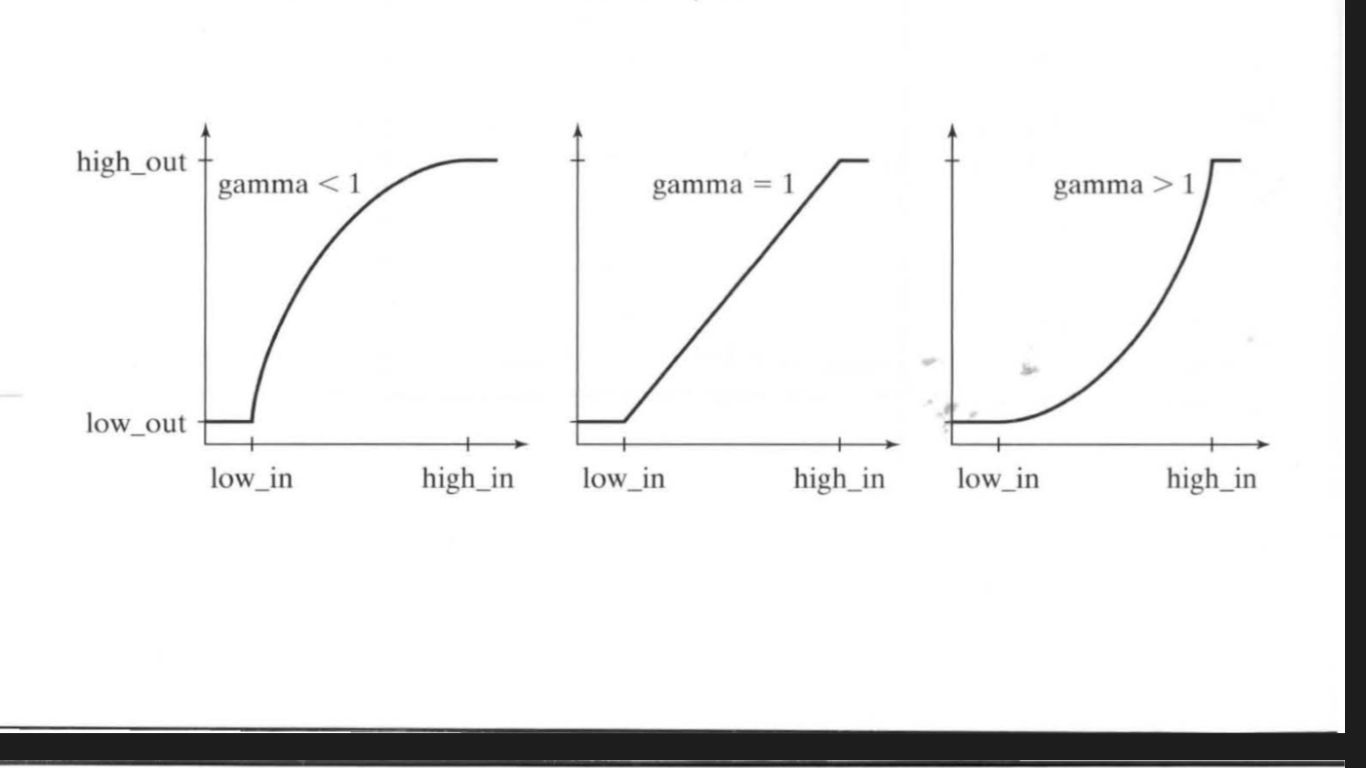
this function maps the intensity values in image f to new values in g

ithe output image and the input image has the same class

All input are specified as values between 0 and 1, independently of the class of f except f and gamma .

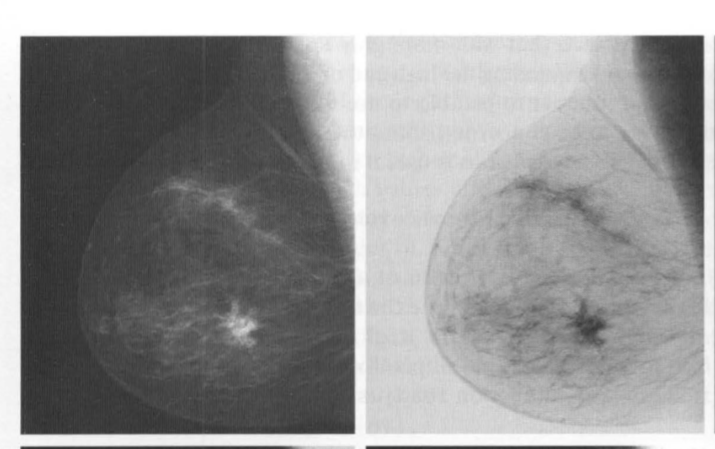
Parameter gamma specifies the shape of the curve that maps the intensity values in f to create g.

* Less than 1 brighter output values
* greater than 1 darker output values
* equal 1 linear mapping (default) ([])



»g1 = imadjust(f, [01], [10]);

This process is particularly useful for enhancing white or gray detail embedded in a large,such as analyze the breast tisue



Sometimes, it is of interest to be able to use function imadjust "automatically," without having to be concerned about the low and high parameters.

Function stretchlim is useful in that regard

* basic syntax is

Low\_High = stretchlim(f)

Low\_High can be used to achieve contrast stretching

Logarithmic transformation

basic tools for dynamic range manipulation.

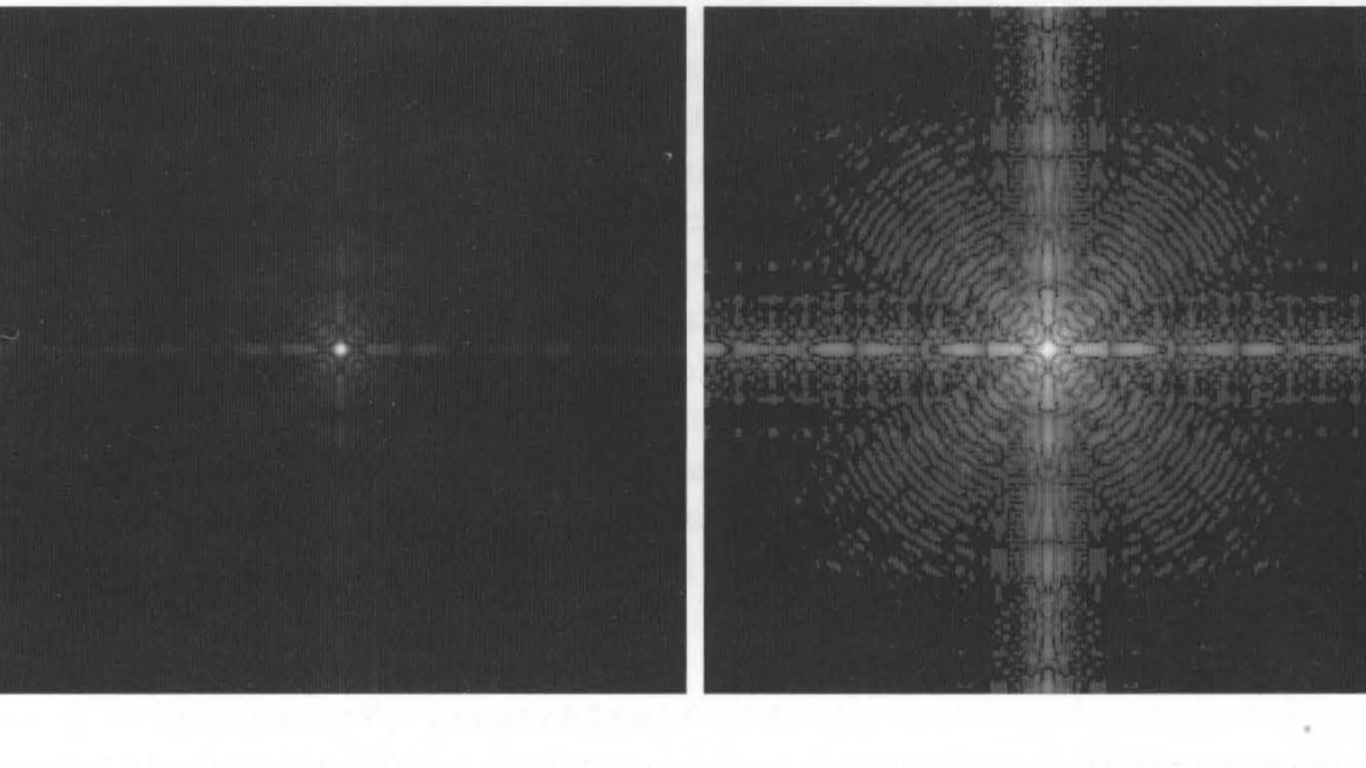
* Logarithm transformations are implemented using the expression

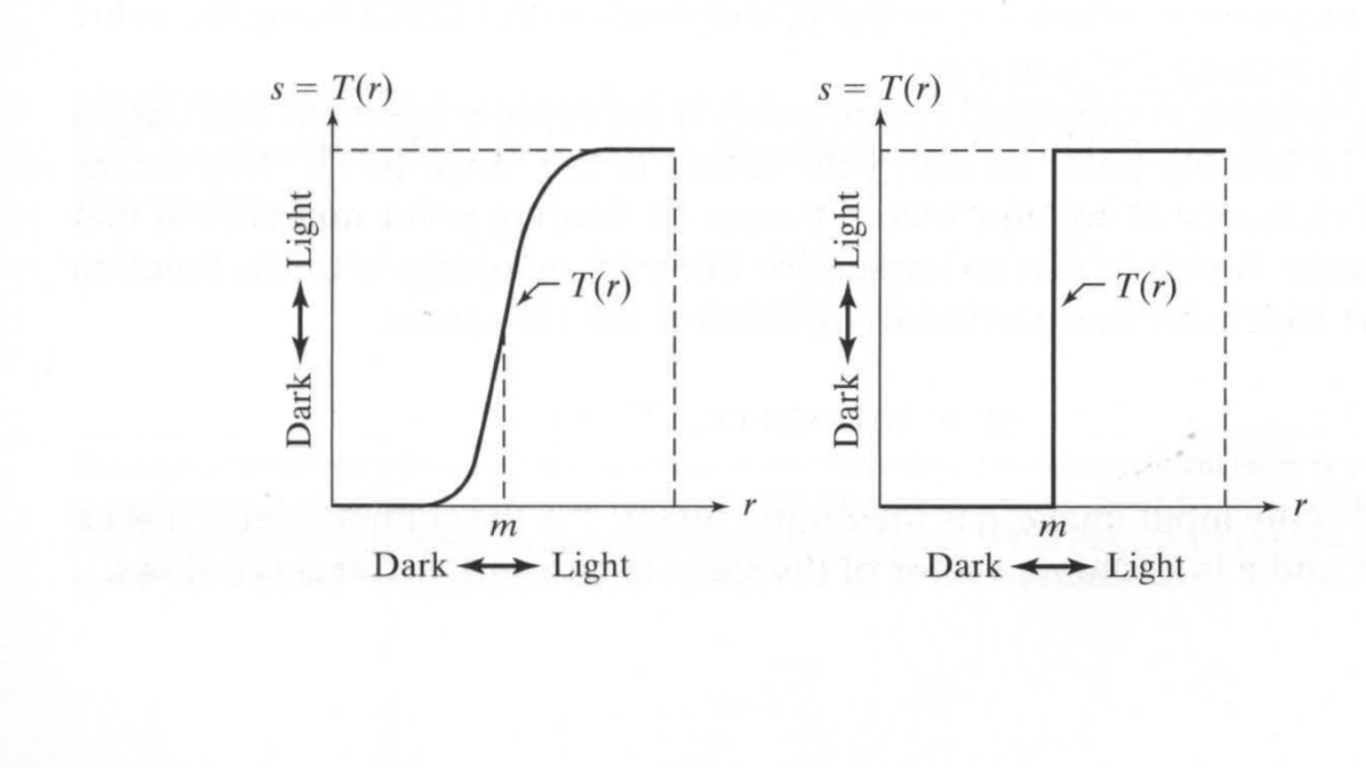
g = c\*log(1 + f)

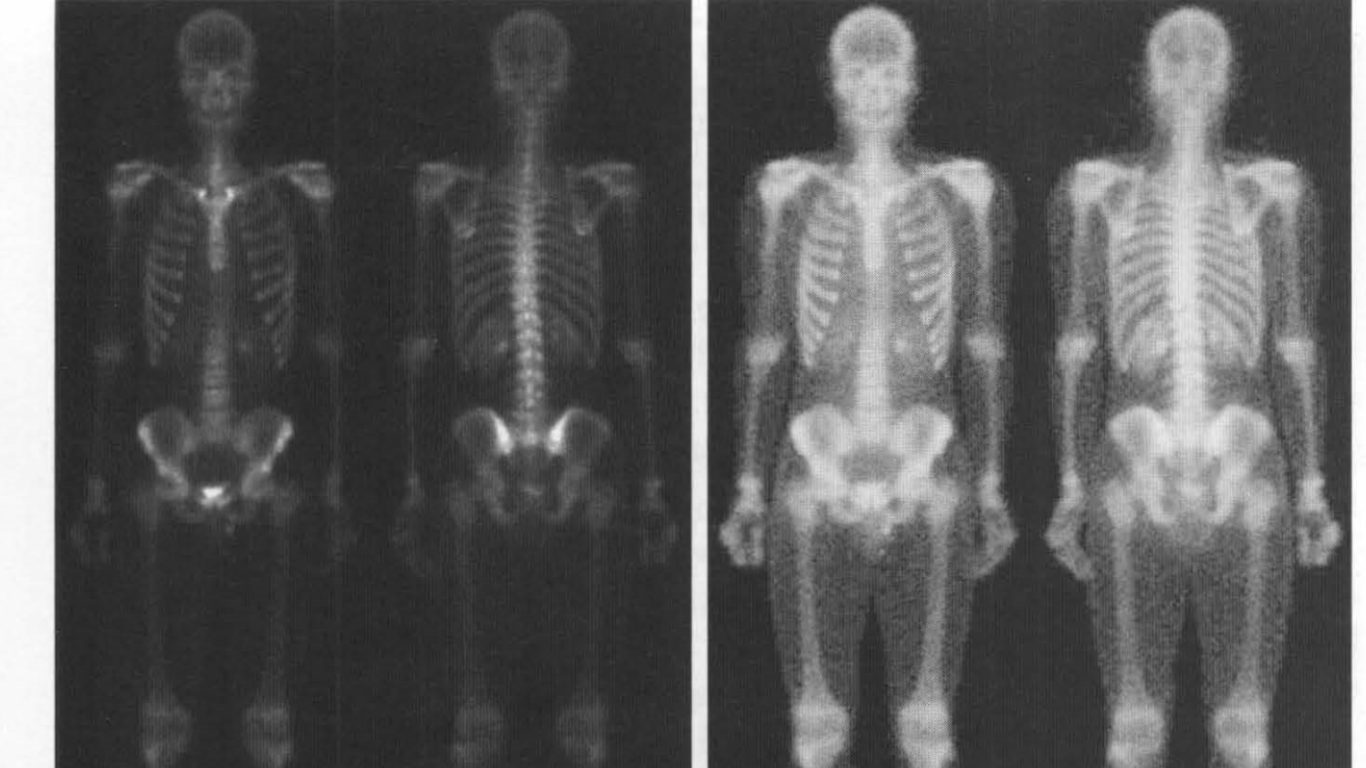
* One of the principal uses of the log transformation is to compress dynamic range.

For example, if we have a Fourier spectrum with values in the range [0,10^6 ] or higher when we display it on the monitor, resulting in lost visual detail in the lower intensity values in the spectrum .

When we use logarithm transformation range will be approximately between 0 to 14



Left contrast stretching function and the right is binary transformation used in segmentaion called thresholding function



Left is the original image and right is the original image after contrast stretching.

# Histogram Processing and Function Plotting

Intensity transformation functions based on information extracted from image intensity histograms play a central role in image processing, in areas such as enhancement, compression, segmentation, and description.

We concentrate here on image enhancement.

Generating and Plotting Image Histograms

• The value of pixel is [0 255] for images of class uint8, [0 65535] for images of class uint16, and [0 1.0] for floating point images.

The simplest way to display the image histogram is

>> imhist(g);

Or specify the number of bins

>>h=imhist(g,b);

. if b = 2 and class type uint8 we divide pixels into to parts [ 0 127] and [128 255]

* We can use bar to plot the histogram

>>H=imhist(g,25);

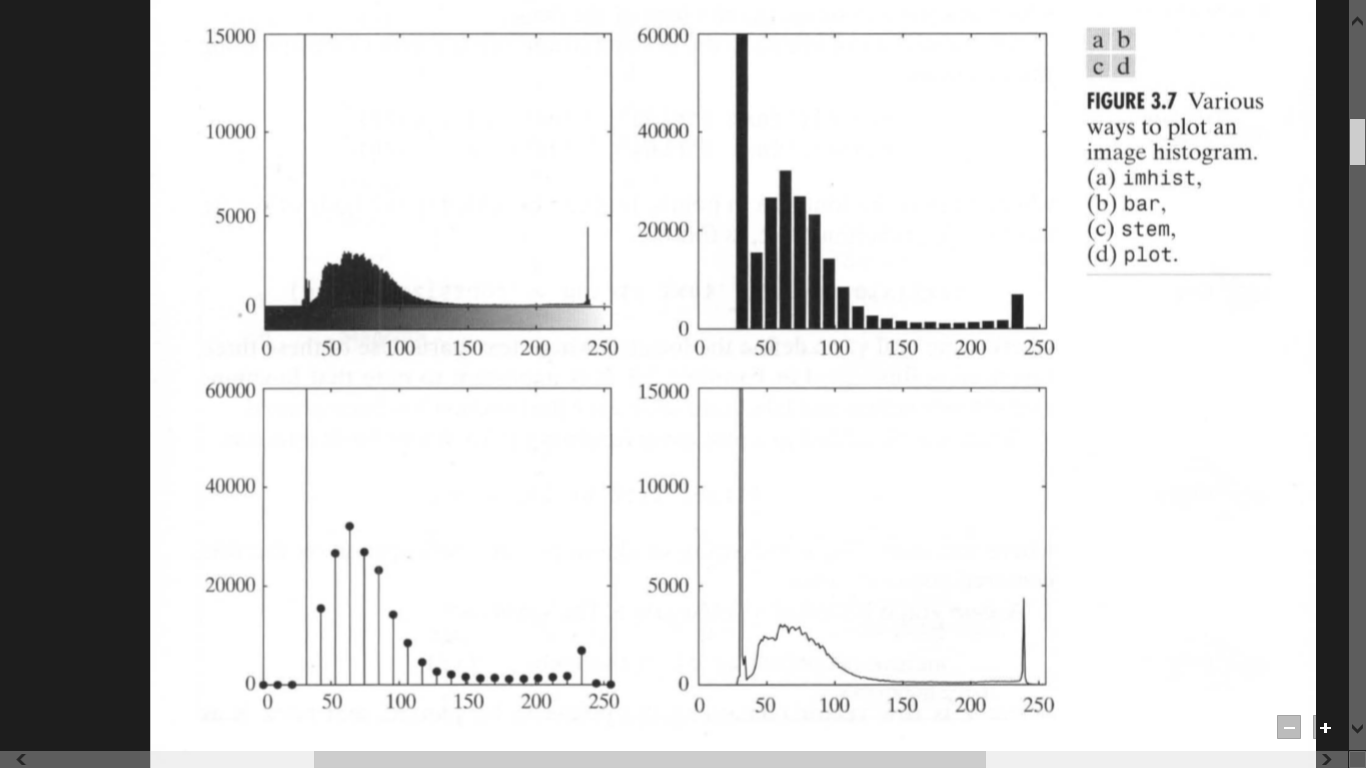
>>horz=linspace(0,255,25);

>>bar(horz,H);

we can also use stem and plot functions

>>bar(horz,H);

>>plot(horz,h,’linespec’);

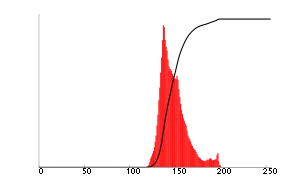


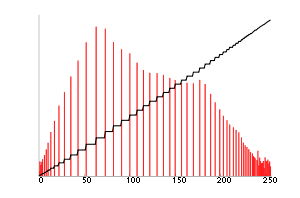
Histogram Equalization

It is a technique of image enhancement where it increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast.

Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

 the method can lead to better views of bone structure in x-rays images, and to better detail in photographs.





Histogram Equalization in IP toolbox

g = histeq (f, nlev)

where f is the image

nlev is the number of intensity levels specified for the output image. nlev = 64 is the default value

An example about histogram equalization

* This image magnified 700 times and has low DR and the most important feature of it is that it is dark.you can deduce that from the histogram
* After using histogram equalization function we have HDR and overcome the darkness feature
* The improvements in average intensity and contrast are evident in the second image.

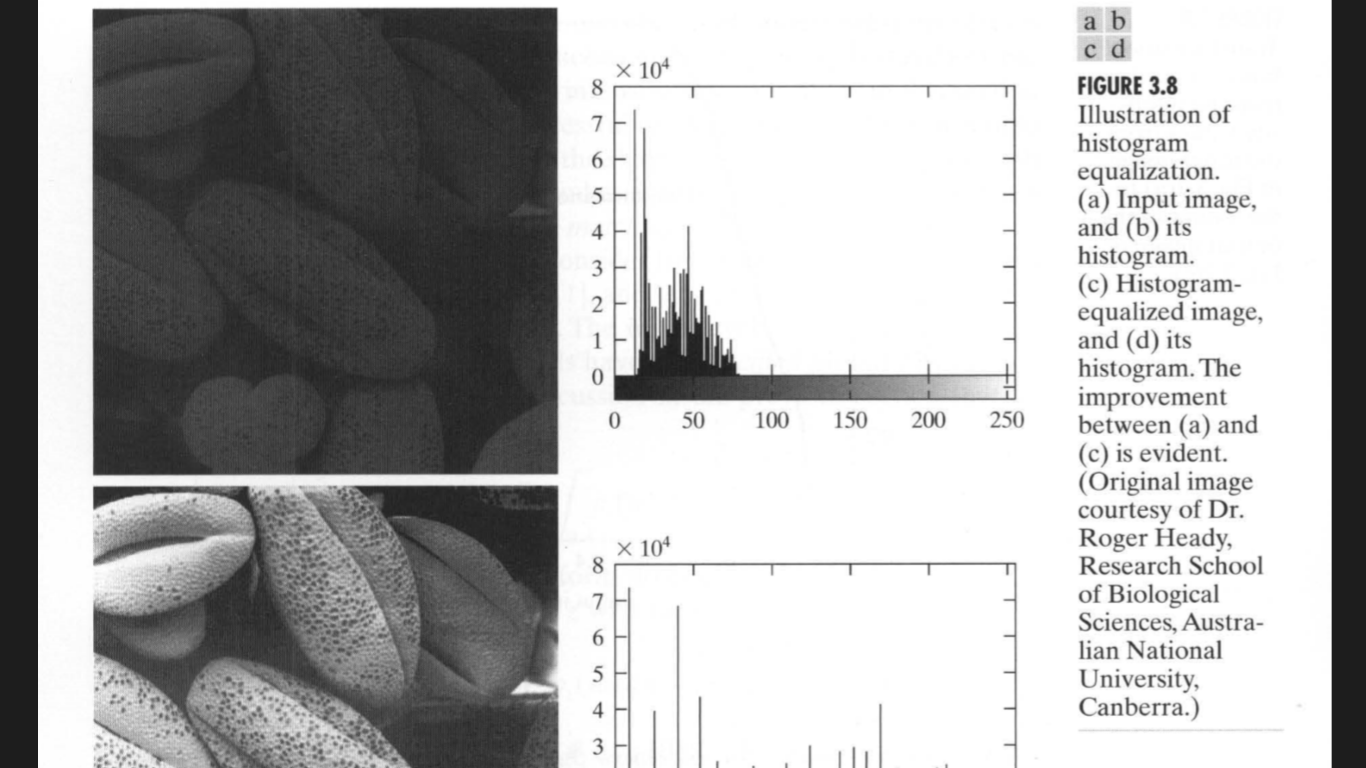


image restoration and reconstruction

Restoration is a process that attempts to recover an image that has been degraded by using a prior knowledge of degradation phenomenon.

Restoration based on modeling the degradation using a prior knowledge and then apply the inverse process in order to restore the original image

This approach usually involves formulating a criterion of goodness that yields an optimal estimate of the desired result. psychophysical aspects of the human visual system is the criterion.

so contast stretch is image enhancement but removal of image blur by deblurring functions is image restoration technique.

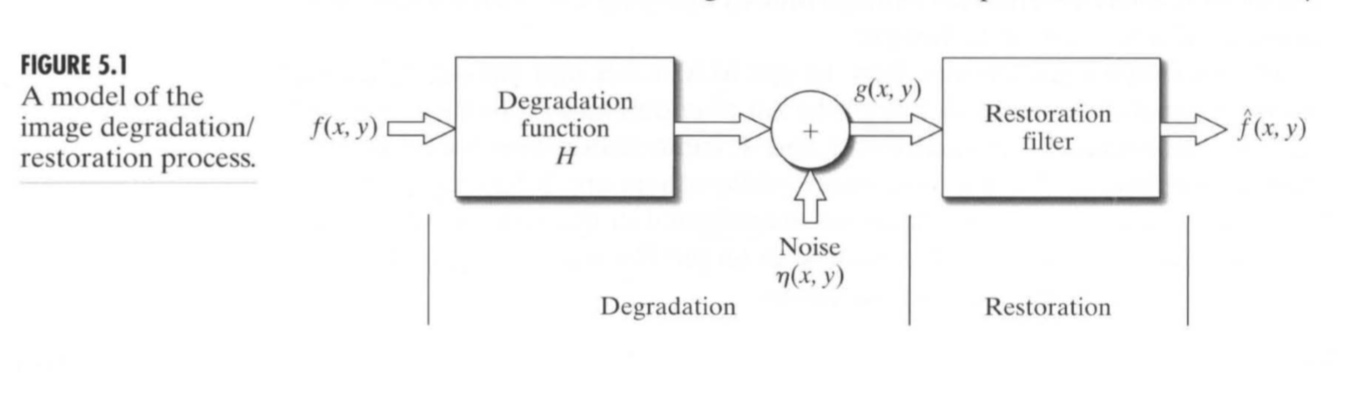
A Model of the Image Degradation/Restoration Process

the degradation process is modeled as a degradation function that, together with an additive noise term, operates on an input image f(x, y) to produce a degraded image g(x, y):

the degraded image is given in the spatial domain by

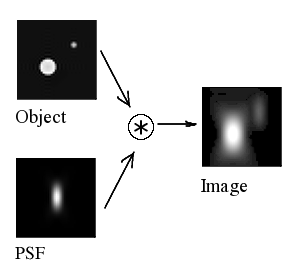
g(x,y) = h(x,y)\* f(x,y) + ᶮ(x,y)

where h(x,y) is the spatial representation of the degradation function



PSF

The **point spread function** describes the response of an imaging system to a [point source](https://en.wikipedia.org/wiki/Point_source) or point object.



**Deblurring an image algrithms**

1. by using **Lucy-Richardson Algorithm :** Use the deconvlucy function to deblur an image
2. by using **the Blind Deconvolution Algorithm:** Use the deconvblind function to deblur an image
3. by using the weiner filter: using the deconvwnr
4. **the Blind Deconvolution Algorithm**

**the Blind Deconvolution Algorithm**

The blind deconvolution algorithm can be used effectively when no information about the distortion (blurring and noise) is known.

* + - * Modeling the Degradation phase
* Create a blurred image by convolving a motion filter PSF with an image.
* I = imread('cameraman.tif');
* figure; imshow(I); title('Original Image');
* PSF= fspecial('motion',13,45);
* % Create the PSF
* figure;imshow(PSF);title('TruePSF')
* B = imfilter(I,PSF,'circ','conv');
* % Simulate the blurred image
* %circ remove the edge related
* %ringing
* figure; imshow(Blurred); title('Blurred Image');

Already, this algorithm is based on trial and error

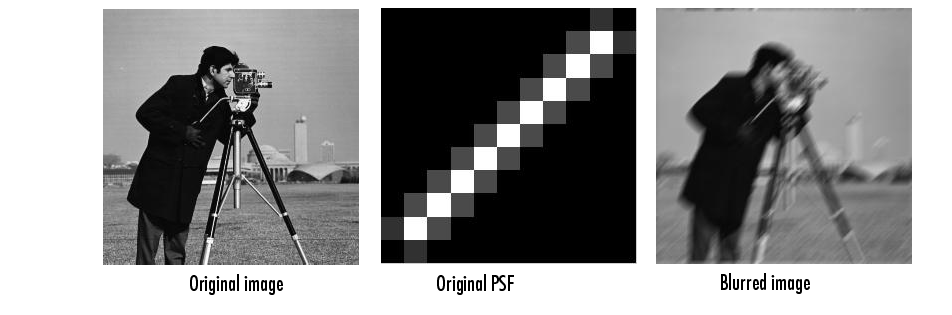
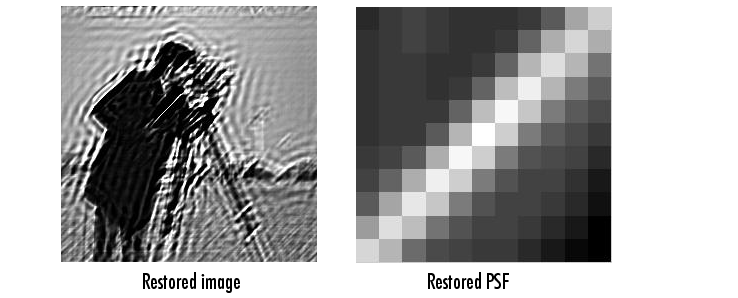


Image restoration

* + - * first step : using deconvblind function specifying the image and an initial guess at the PSF as arguments.
* INITPSF = ones(size(PSF));
* [J P]= deconvblind(Blurred,INITPSF,30);
* figure; imshow(J); title('Preliminary Restoration');
* figure; imshow(P,[],'notruesize');
* title('Preliminary Restoration');



the edge-related ringing in the restored image around the sharp intensity contrast areas is unsatisfactory.

So we use weight array to exclude areas of high-contrast from the deblurring operation.

This example uses edge detection and morphological processing to create a WEIGHT array.

The edge, strel, and imdilate functions to detect the high-contrast areas in the image.

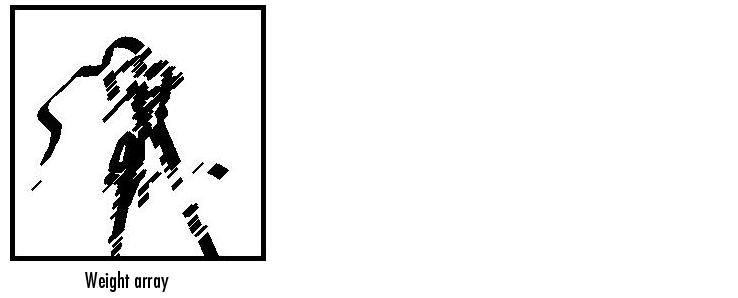
WEIGHT = edge(I,'sobel',.28);

se1 = strel('disk',1); %strel create element structure that can be used in morphological

se2 = strel('line',13,45);

WEIGHT = imdilate(WEIGHT,[se1 se2]);

figure; imshow(WEIGHT); title('Weight array');



[J2 P2] = deconvblind(Blurred,P1,50,[],WEIGHT);

figure; imshow(J2);

title('Newly deblurred image');

figure; imshow(P2);

title('Newly reconstructed PSF');

