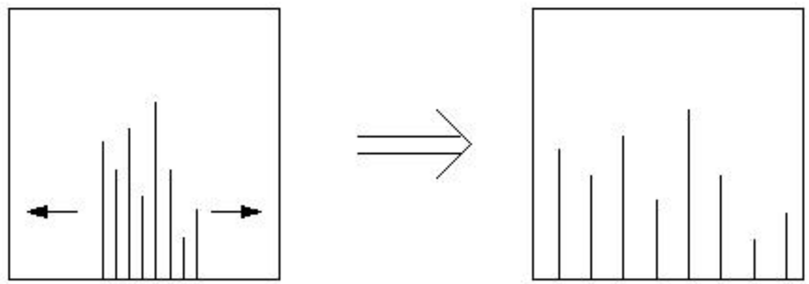
IMAGE ENHANCEMENT

Histogram Modification

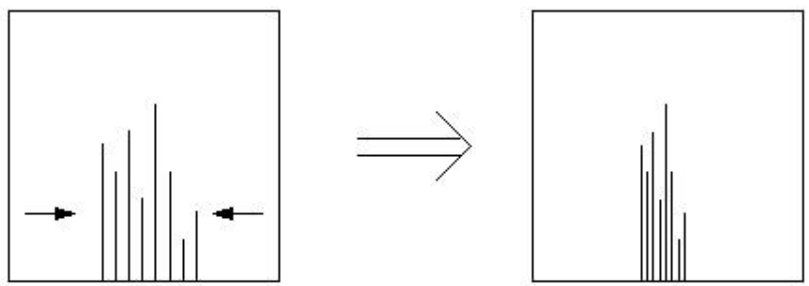
- ✔ Histogram modification performs a function similar to gray level mapping, but works by considering histogram's shape and spread
- ✔ ***Gray level histogram*** of an image is the distribution of the gray levels in an image
- ✔ Examination of the histogram is one of the most useful tools for image enhancement, as it makes easy to see the modifications that may improve an image

- The histogram can be modified by a mapping function, which will **stretch**, **shrink** (compress), or **slide** the histogram
- Histogram stretching and histogram shrinking are forms of gray scale modification, sometimes referred to as ***histogram scaling***

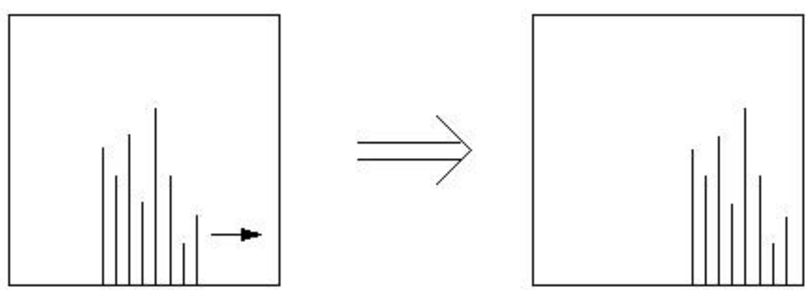
Figure 8.2-8: Histogram Modification



a) Histogram stretch



b) Histogram shrink



c) Histogram slide

✔ *Histogram stretch*

- The mapping function equation is as follows:

$$Stretch(I(r,c)) = \left[\frac{I(r,c) - I(r,c)_{MIN}}{I(r,c)_{MAX} - I(r,c)_{MIN}} \right] [MAX - MIN] + MIN$$

where: $I(r,c)_{MAX}$ is the largest gray level value in the image $I(r,c)$, $I(r,c)_{MIN}$ is the smallest gray level value in $I(r,c)$ and

MAX and MIN correspond to the maximum and minimum gray level values possible (for an 8-bit image these are 0 and 255)

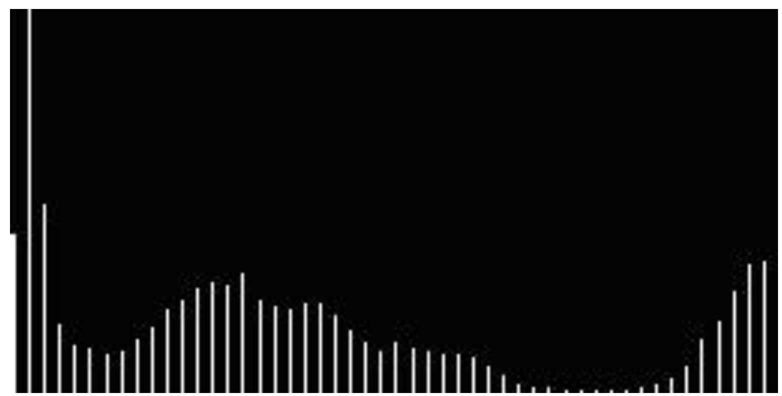
- This equation will take an image and stretch the histogram across the entire gray level range, which has the effect of increasing the contrast of a low contrast image
- If most of the pixel values in an image fall within a small range, it is useful to allow a small percentage of the pixel values to be clipped at the low and high end of the range (for an 8-bit image this means truncating at 0 and 255)

Histogram Stretching

a) Low-contrast image

b) Histogram of image (a)

c) Image (a) after histogram stretch



d) Histogram of image after stretch

Histogram Stretching with Clipping

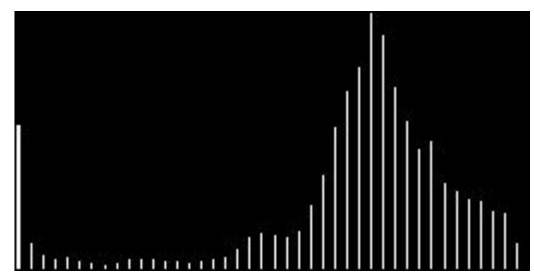
a) Original image

b) Histogram of original image

c) Image after histogram stretching with out clipping

d) Histogram of image (c)

e) Image after histogram stretching with clipping 1% of the values at the high and low ends



f) Histogram of image (e)

- ***Histogram shrink***
- The mapping function equation is as follows:

$$Shrink(I(r,c)) = \left[\frac{Shrink_{MAX} - Shrink_{MIN}}{I(r,c)_{MAX} - I(r,c)_{MIN}}\right]\left[I(r,c) - I(r,c)_{MIN}\right] + Shrink_{MIN}$$

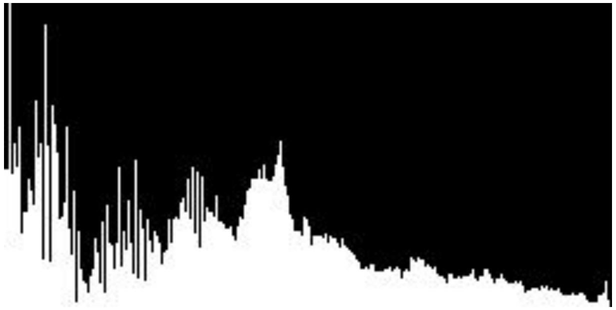
the image I(r,c), $I(r,c)_{MIN}$ is the smallest gray level value in I(r,c) and

$Shrink_{MAX}$ and $Shrink_{MIN}$ correspond to the maximum and minimum desired in the compressed histogram

- Decreases image contrast by compressing the gray levels
- However this method may not be useful as an image enhancement tool, but it is used in an image sharpening algorithm (unsharp masking) as a part of an enhancement technique

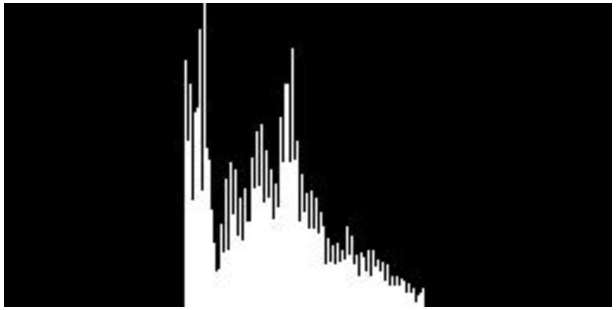
Histogram Shrinking

a) Original image



b) Histogram of image

c) Image after shrinking the histogram to the range [75,175]



d) Histogram of image (c)

✔ *Histogram slide*

- Used to make an image either darker or lighter, but retain the relationship between gray level values
- Accomplished by simply adding or subtracting a fixed number from all of the gray level values, as follows:

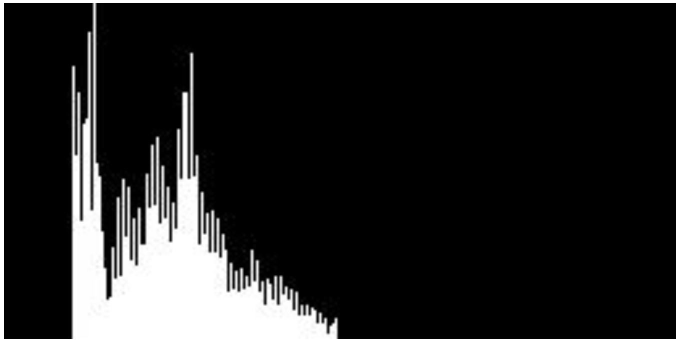
$$Slide(I(r, c)) = I(r,c) + OFFSET$$

where the OFFSET value is the amount to slide the histogram

- In this equation we assume that any values slid past the minimum and maximum values will be clipped to the respective minimum or maximum
- A positive OFFSET value will increase the overall brightness, while a negative OFFSET will create a darker image

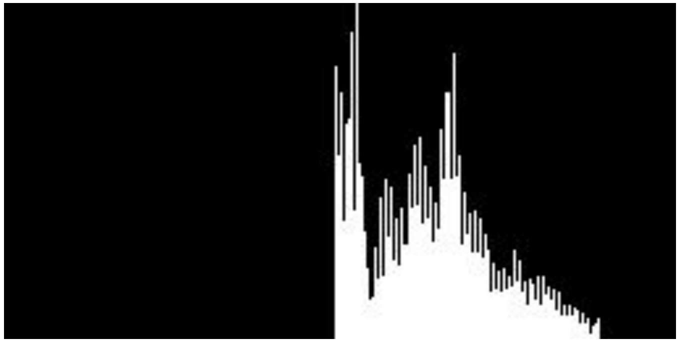
Histogram Slide

a) Resultant image from sliding the histogram down by 50



b) Histogram of image (a)

c) Resultant image from sliding the histogram up by 50



d) Histogram of image (c)

✔ *Histogram equalization*

- A technique where the histogram of the resultant image is as flat as possible
- The theoretical basis for histogram equalization involves probability theory, where we treat the histogram as the probability distribution of the gray levels
- Its function is similar to that of a histogram stretch but often provides more visually pleasing results across a wider range of images

- Consists of four steps:

  1. Find the running sum of the histogram values
  2. Normalize the values from step (1) by dividing by the total number of pixels
  3. Multiply the values from step (2) by the maximum gray level value and round
  4. Map the gray level values to the results from step (3) using a one-to-one correspondence

Example:

3-bits per pixel image – range is 0 to 7.

Given the following histogram:

Gray Level Value	Number of Pixels (Histogram values)
0	10
1	8
2	9
3	2
4	14
5	1
6	5
7	2

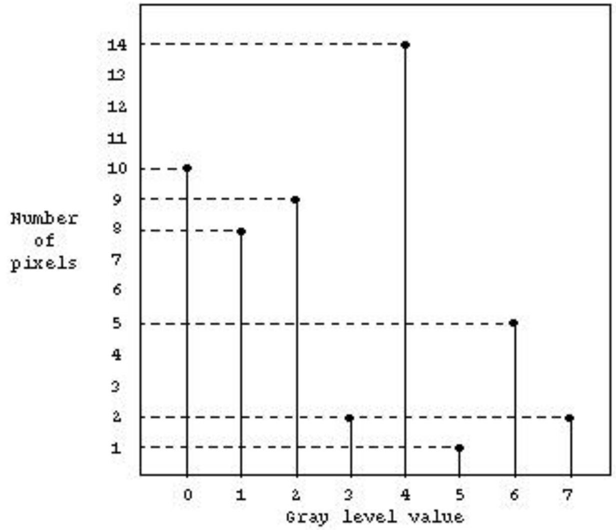
1) Create a running sum of the histogram values. This means the first value is 10, the second is 10+8=18, next 10+8+9=27, and so on. Here we get 10, 18, 27, 29, 43, 44, 49, 51
2) Normalize by dividing by the total number of pixels. The total number of pixels is: 10+8+9+2+14+1+5+0 = 51 (note this is the last number from step 1), so we get: 10/51, 18/51, 27/51, 29/51, 43/51, 44/51, 49/51, 51/51
3) Multiply these values by the maximum gray level values, in this case 7, and then round the result to the closest integer. After this is done we obtain: 1, 2, 4, 4, 6, 6, 7, 7

4) Map the original values to the results from step 3 by a one-to-one correspondence. This is done as follows:

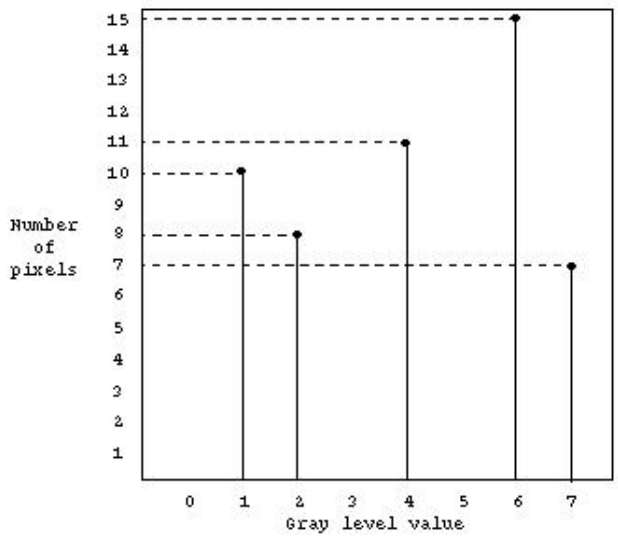
Original Gray Level Value	Histogram Equalized Values
0	1
1	2
2	4
3	4
4	6
5	6
6	7
7	7

- All pixels in the original image with gray level 0 are set to 1, values of 1 are set to 2, 2 set to 4, 3 set to 4, and so on. After the histogram equalization values are calculated and can be implemented efficiently with a look-up-table (LUT), as discussed in Chapter 2
- We can see the original histogram and the resulting histogram equalized histogram in Fig. 8.2.14. Although the result is not flat, it is closer to being flat than the original histogram

Figure 8.2-14: Histogram Equalization



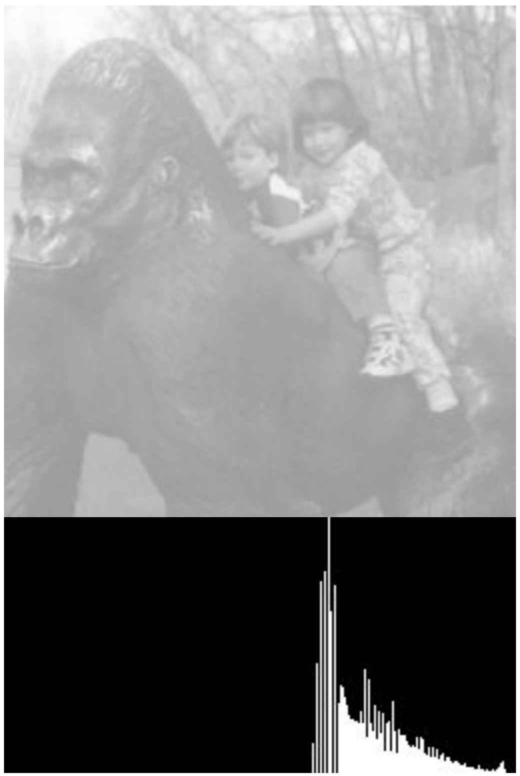
a) Original histogram



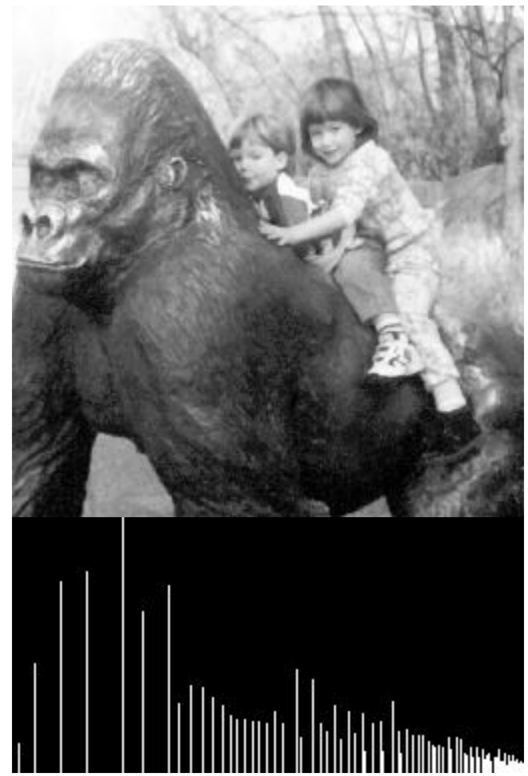
b) After histogram equalization

Histogram Equalization Examples

1.



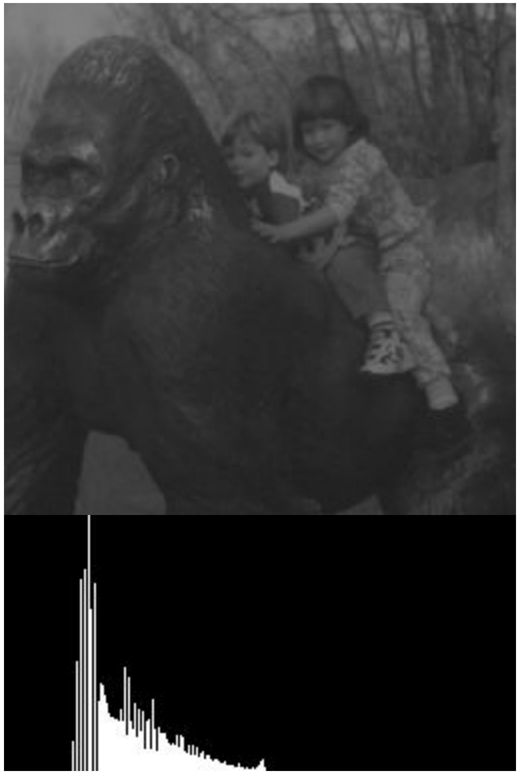
Input image



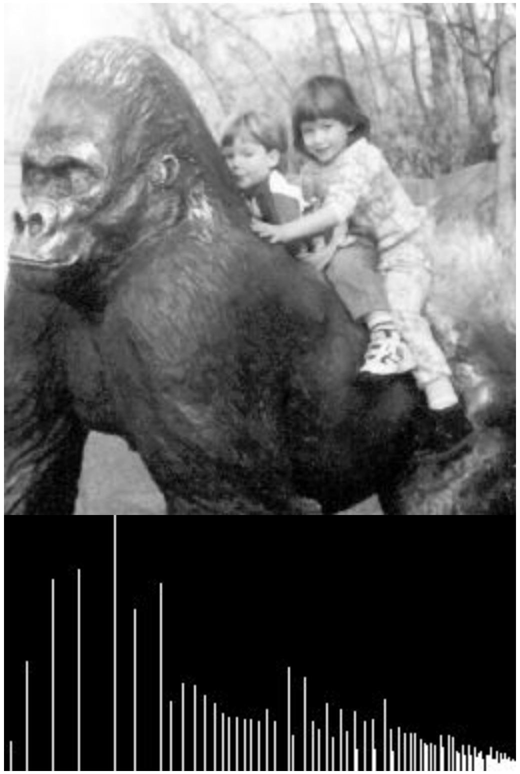
Resultant image after histogram equalization

Histogram Equalization Examples (contd)

2.



Input image



Resultant image after histogram equalization

Note: As can be seen histogram equalization provides similar results regardless of the input image

- Histogram equalization of a digital image will not typically provide a histogram that is perfectly flat, but it will make it as flat as possible
- Histogram equalization may not always provide the desired effect, since its goal is fixed – to distribute the gray level values as evenly as possible. To allow for interactive histogram manipulation, the ability to specify the histogram is necessary

IMAGE ENHANCEMENT

SPATIAL AVERAGING

Image Averaging

- A noisy image:

$$g(x, y) = f(x, y) + n(x, y)$$

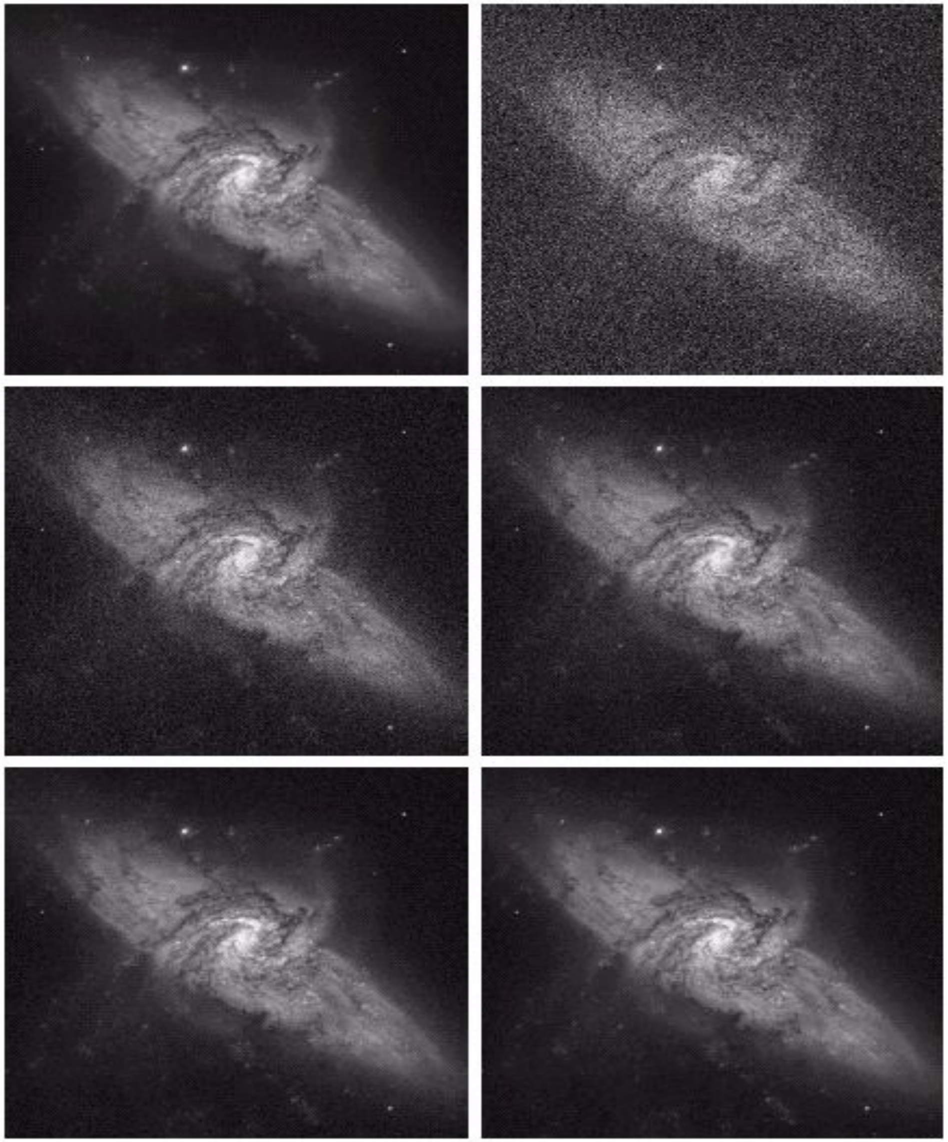
- Averaging K different noisy images:

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

Image Averaging

- As K increases, the variability of the pixel values at each location decreases.
  - This means that $\bar{g}(x,y)$ approaches $f(x,y)$ as the number of noisy images used in the averaging process increases.
- Registering(aligned) of the images is necessary to avoid blurring in the output image.

Image Enhancement in the Spatial Domain

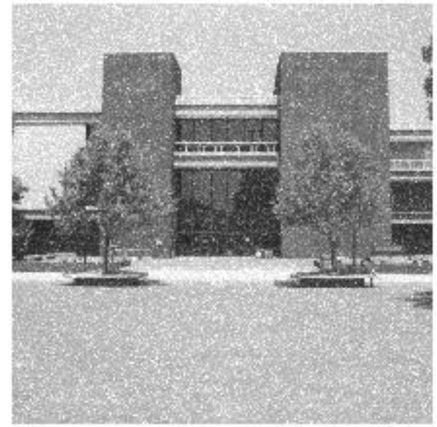


a b
c d
e f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (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. (Original image courtesy of NASA.)

Figure 9.3-6: Arithmetic Mean Filter

a) Image with gaussian noise
variance = 300, mean = 0



b) Image with gamma noise
variance = 300, alpha = 1

c) Result of arithmetic mean
filter, mask size = 3, on
image with gaussian noise

d) Result of arithmetic mean
filter, mask size = 3, on
image with gamma noise

Figure 9.3.6, continued

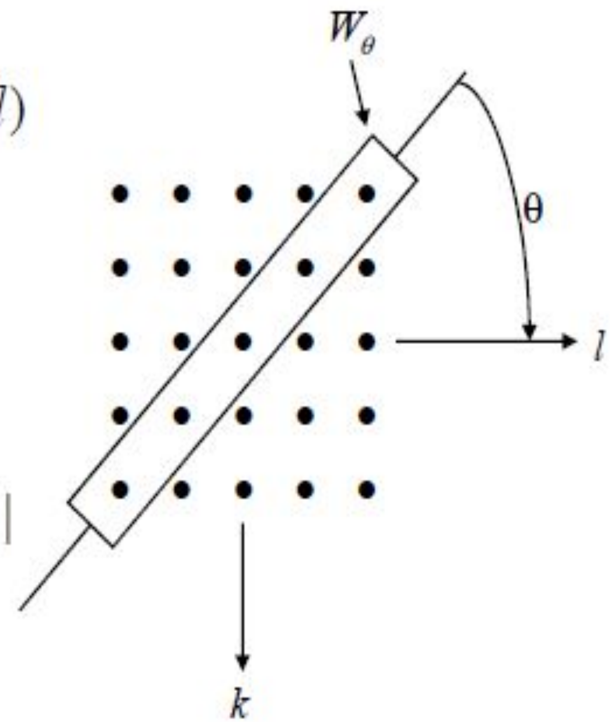
e) Result of arithmetic mean filter, mask size = 5, on image with gaussian noise

f) Result of arithmetic mean filter, mask size = 5, on image with gamma noise

- Directional Smoothing
  - to protect the edges from blurring while smoothing

$$v(m,n:\theta)=\frac{1}{N_\theta}\sum_{(k,l)\in W_\theta}\sum y(m-k,n-l)$$

$$\Rightarrow v(m,n)=v(m,n:\theta^*)$$



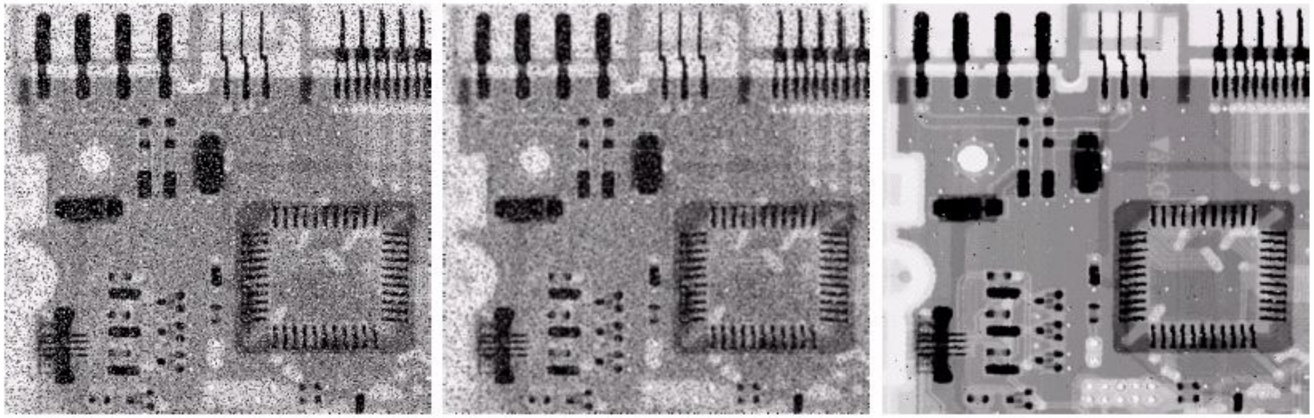
Find out θ^*

such that $\min|\,y(m,n)-v(m,n:\theta^*)\,|$

Smoothing Filters

- Median filtering (nonlinear)
  - Used primarily for noise reduction (eliminates isolated spikes)
  - The gray level of each pixel is replaced by the median of the gray levels in the neighborhood of that pixel (instead of by the average as before).

Chapter 3
Image Enhancement in the Spatial Domain



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.)

Median Filter

a) Image with added salt-and-pepper noise, the probability for salt = probability for pepper = 0.10

b) After median filtering with a 3x3 window,all the noise is not removed

Median Filter

c) After median filtering with a 5x5 window, all the noise is removed, but the image is blurry acquiring the “painted” effect

- The ***contra-harmonic mean*** *filter* works well for images containing salt OR pepper type noise, depending on the filter order, *R*:

$$\text{Contra - Harmonic Mean} = \frac{\sum_{(r,c)\in W} d(r,c)^{R+1}}{\sum_{(r,c)\in W} d(r,c)^{R}}$$

where W is the NxN window under consideration

- For negative values of R, it eliminates salt-type noise, while for positive values of R, it eliminates pepper-type noise

Figure 9.3.9: Contra-harmonic Mean Filter

a) Image with salt noise, probability = .04

b) Result of contra-harmonic mean filter, mask size = 3, order = -3

c) Image with pepper noise, probability = .04

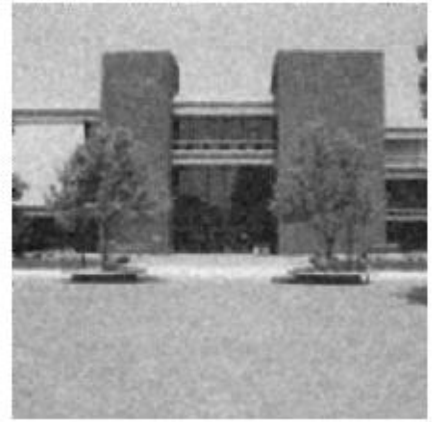
d) Result of contra-harmonic mean filter, mask size = 3, order = +3

- The ***geometric mean filter*** works best with Gaussian noise, and retains detail information better than an arithmetic mean filter
- It is defined as the product of the pixel values within the window, raised to the 1/(N*N) power:

$$\text{Geometric Mean} = \prod_{(r,c) \in W} [d(r,c)]^{\frac{1}{N^2}}$$

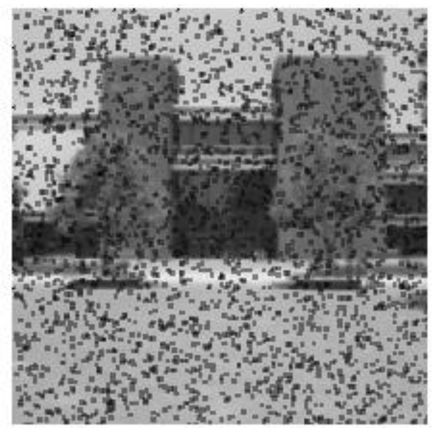
Figure 9.3-10: Geometric Mean Filter

a) Image with gaussian noise, variance = 300, mean = 0



b) Result of geometric mean filter, mask size = 3, on image with gaussian noise

c) Image with pepper noise, probability = .04



d) Result of geometric mean filter, mask size = 3, on image with pepper noise

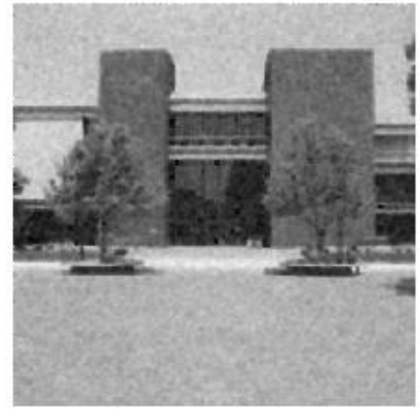
- The ***harmonic mean filter*** fails with pepper noise, but works well for salt noise
- It is defined as follows:

$$Harmonic\ Mean = \frac{N^2}{\sum_{(r,c)\in W} \frac{1}{d(r,c)}}$$

- This filter also works with Gaussian noise, retaining detail information better than the arithmetic mean filter

Figure 9.3-11: Harmonic Mean Filter

a) Image with gaussian noise, variance = 300, mean = 0



b) Result of harmonic mean filter, mask size = 3, on image with gaussian noise

c) Image with salt noise, probability = .04

d) Result of harmonic mean filter, mask size = 3, on image with salt noise

- The *Yp mean filter* is defined as follows:

$$Y_p\,Mean = \left[\sum_{(r,c)\in W} \frac{d(r,c)^P}{N^2}\right]^{\frac{1}{P}}$$

- This filter removes salt noise for negative values of *P*, and pepper noise for positive values of *P*

Figure 9.3-12: Yp Mean Filter

a) Image with salt noise, probability = .04

b) Result of Yp mean filter, mask size = 3, order = -3 on image with salt noise

c) Image with pepper noise, probability = .04

d) Result of Yp mean filter, mask size = 3, order = +3 on image with pepper noise

HOMOMORPHIC FILTERING

- It simultaneously normalizes the brightness across an image and increases contrast.
- Filtering is used to remove multiplicative_noise
- Illumination and reflectance are not separable
- Illumination and reflectance combine multiplicatively
- Components are made additive by taking the logarithm of the image intensity

- Multiplicative components of the image can be separated linearly in the frequency domain

- To make the illumination of an image more even, the high-frequency components are increased and low-frequency components are decreased

- High-frequency components are assumed to represent mostly the reflectance in the scene (the amount of light reflected off the object in the scene)

- Low-frequency components are assumed to represent mostly the illumination in the scene

- High-pass_filtering is used to suppress low frequencies and amplify high frequencies, in the log-intensity domain

- The illumination component tends to vary slowly across the image.

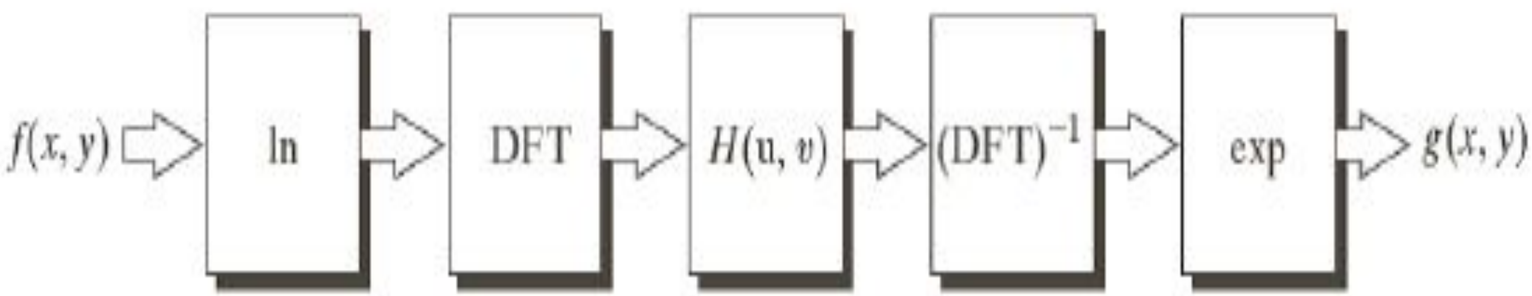
- The reflectance tends to vary rapidly, particularly at junctions of dissimilar objects.

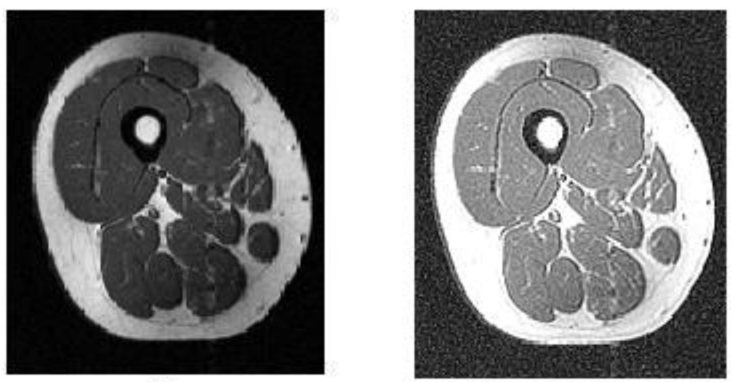
- Therefore, by applying a frequency domain filter of the form we can reduce intensity variation across the image while highlighting detail.

$$f(x,y) = i(x,y) \cdot r(x,y)$$

$$g = \ln f = \ln i + \ln r.$$

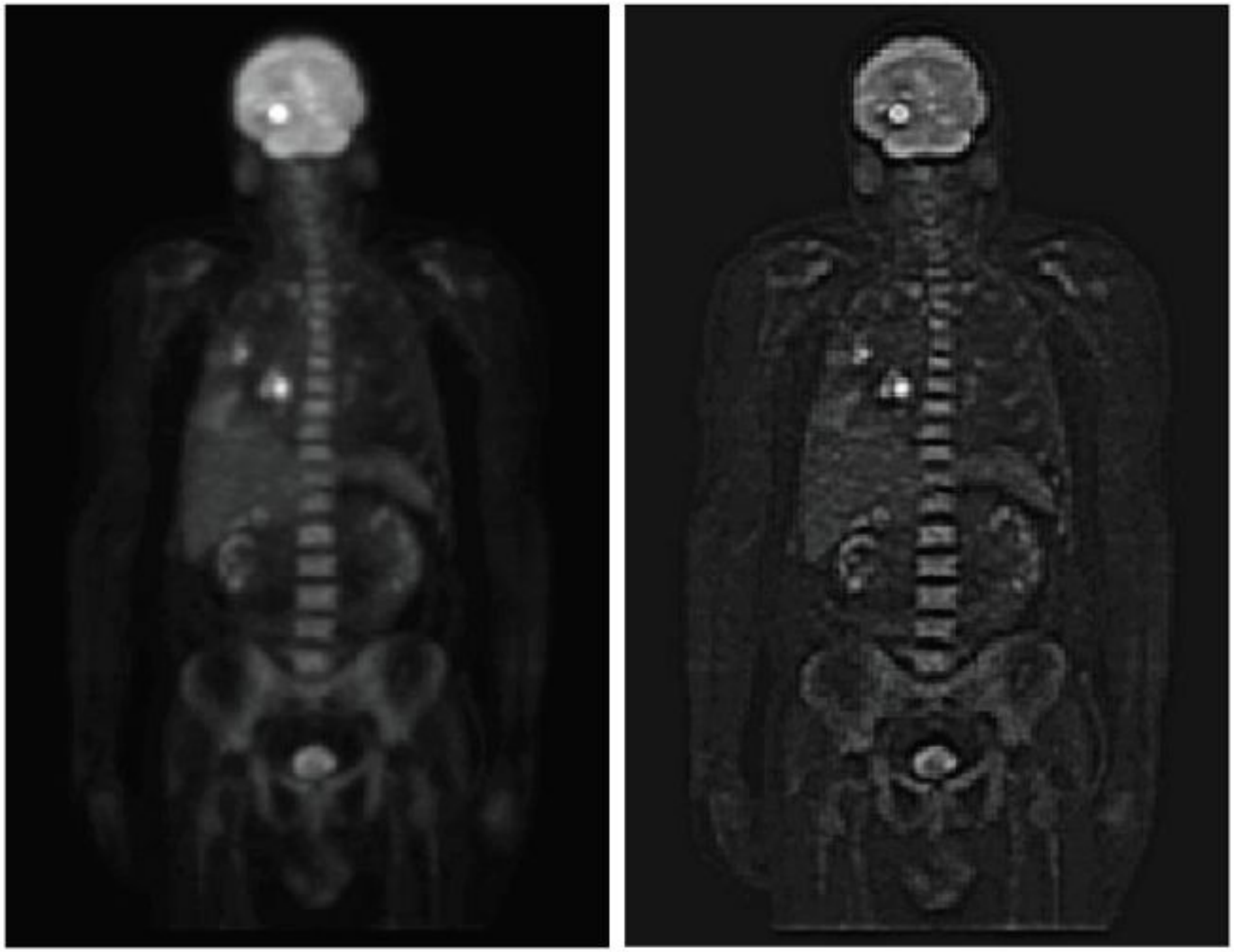
$$\mathcal{F}\{g(x,y)\} = \mathcal{F}\{\ln i(x,y)\} + \mathcal{F}\{\ln r(x,y)\}$$
$$G(u,v) = I_l(u,v) + R_l(u,v).$$





(a) (b)

Homomorphic filtering: PET example



Color
Image Enhancement

Enhancement of Color Images

- Gray scale transforms and histogram modification techniques can be applied by treating a color image as three gray images

- Care must be taken in how this is done to avoid color shifts

- ✔ Histogram modification can be performed on color images, but doing it on each color band separately can create relative color changes

- ✔ The relative color can be retained by applying the gray scale modification technique to one of the color bands, and then using the ratios from the original image to find the other values

Histogram Equalization of Color Images

a) Original poor contrast image

b) Histogram equalization based on the red color band

Histogram Equalization of Color Images (contd)

c) Histogram equalization based on the green color band

d) Histogram equalization based on the blue color band

Note: In this case the red band gives the best results
This will depend on the image and the desired result

- ✔ Typically the most important color band is selected, and this choice is very much application-specific and will not always provide us with the desired result

- ✔ Often, we really want to apply the gray scale modification method to the image brightness only, even with color images

- ✔ Histogram modification on color images can be performed in the following ways:

  - Retain the RGB ratios and perform the modification on one band only, then use the ratios to get the other two bands’ values, or

  - Perform a color transform, such as HSL, do the modification on the lightness (brightness band), then do the inverse color transform