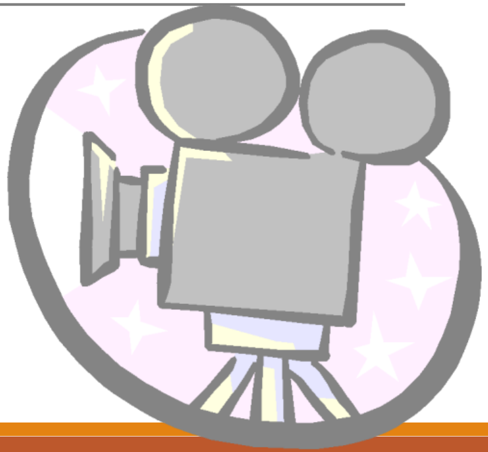
Image Processing

CS-317/CS-341



Outline

- Histogram Processing
  - Local Histogram Processing
- Arithmetic/Logic Operations

Discrete formulation

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

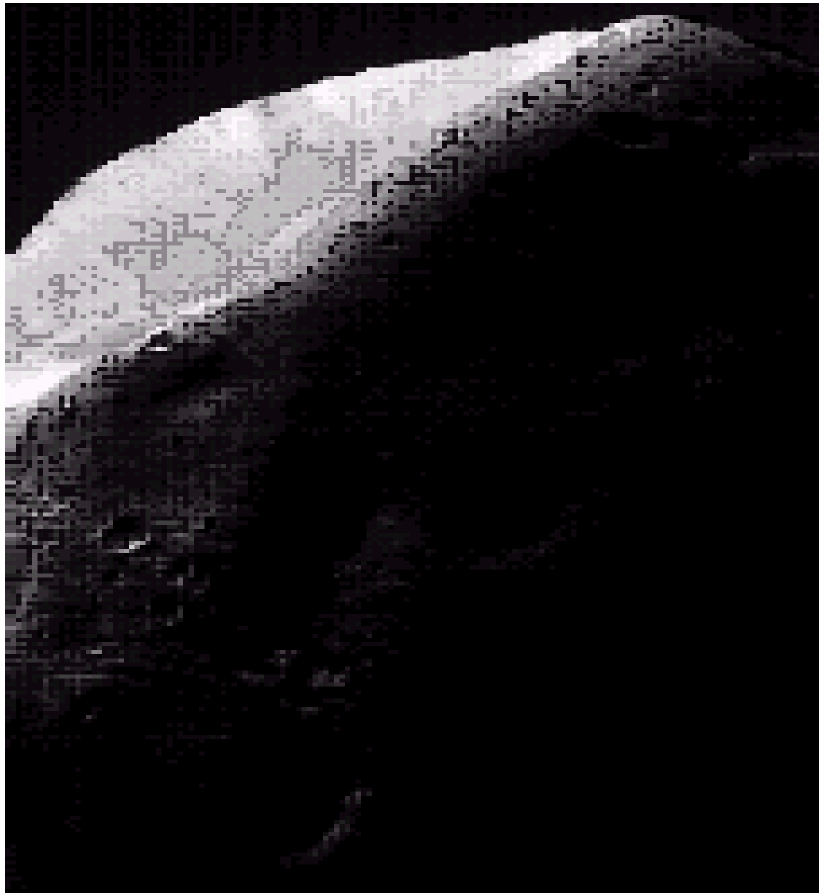
$$= (L-1)\sum_{j=0}^{k} \frac{n_j}{MN} \qquad k = 0,1,2,\ldots,L-1$$

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

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

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

Example



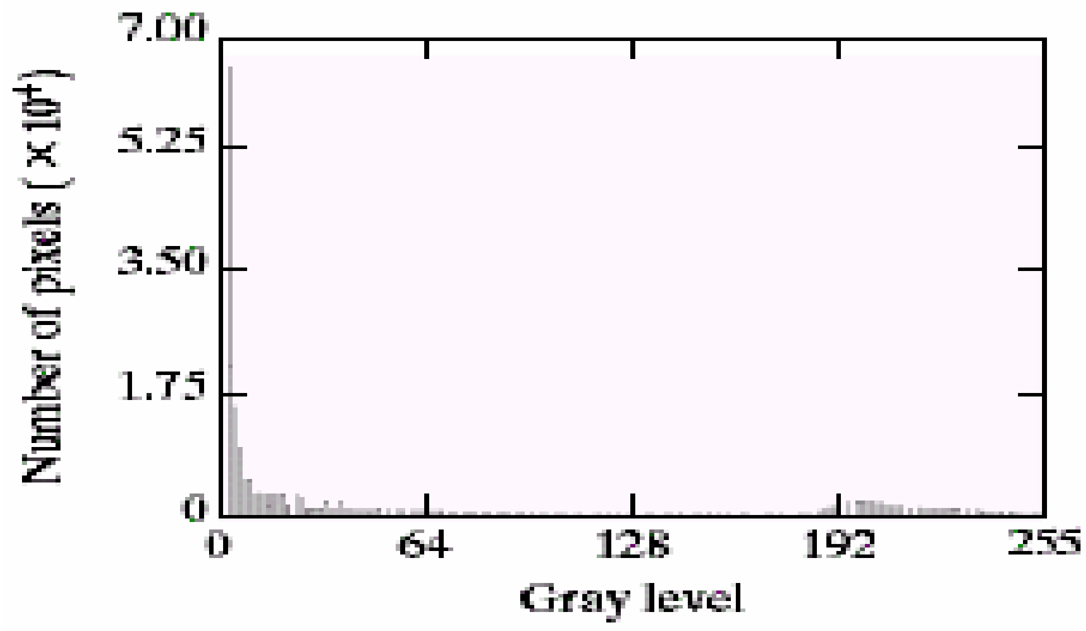
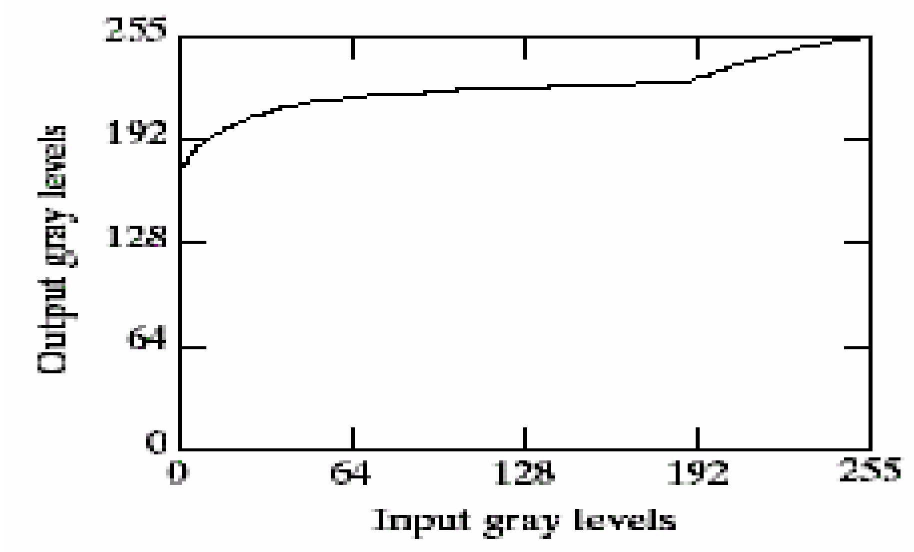


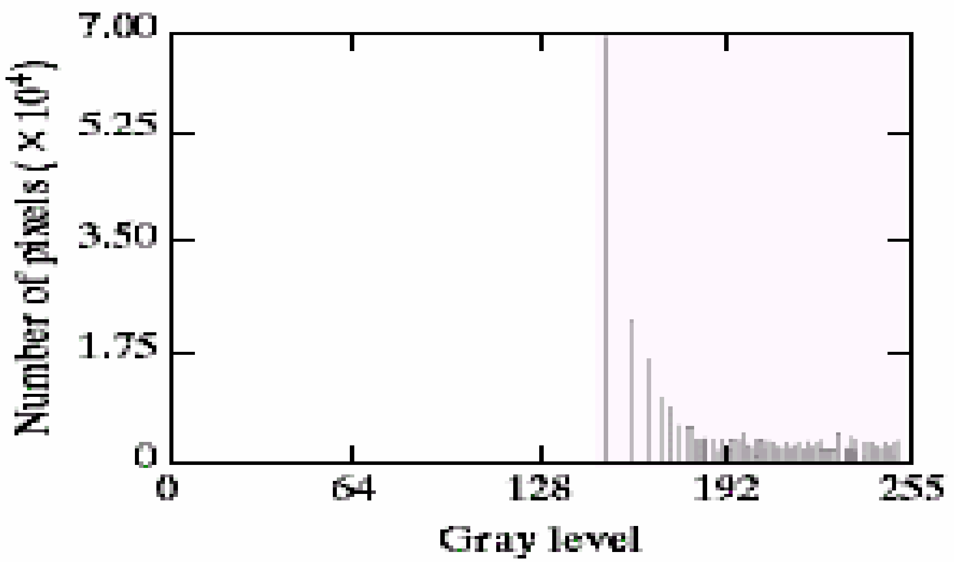
Image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in pixels in the dark end of the gray scale

Image of Mars moon

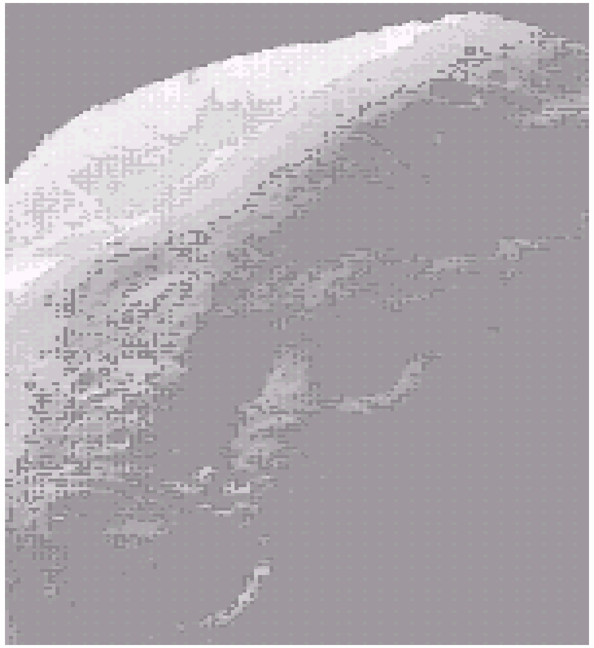
Image Equalization



Transformation function for histogram equalization



Histogram of the result image



Result image after histogram equalization

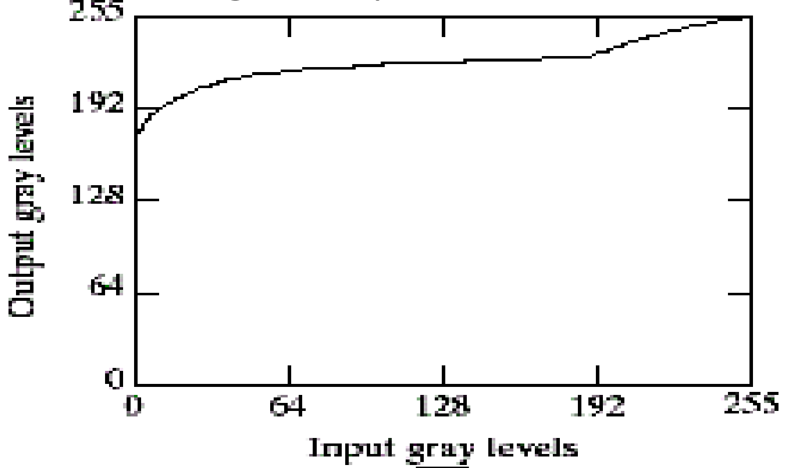
The histogram equalization doesn't make the result image look better than the original image. Consider the histogram of the result image, the net effect of this method is to map a very narrow interval of dark pixels into the upper end of the gray scale of the output image. As a consequence, the output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

Problem with the transformation function and solution

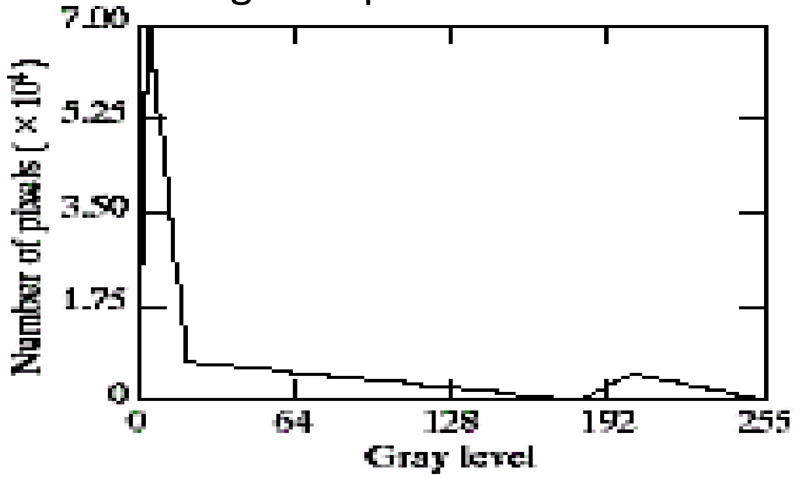
Since the problem with the transformation function of the histogram equalization was caused by a large concentration of pixels in the original image with levels near 0

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

Histogram Equalization



Histogram Specification



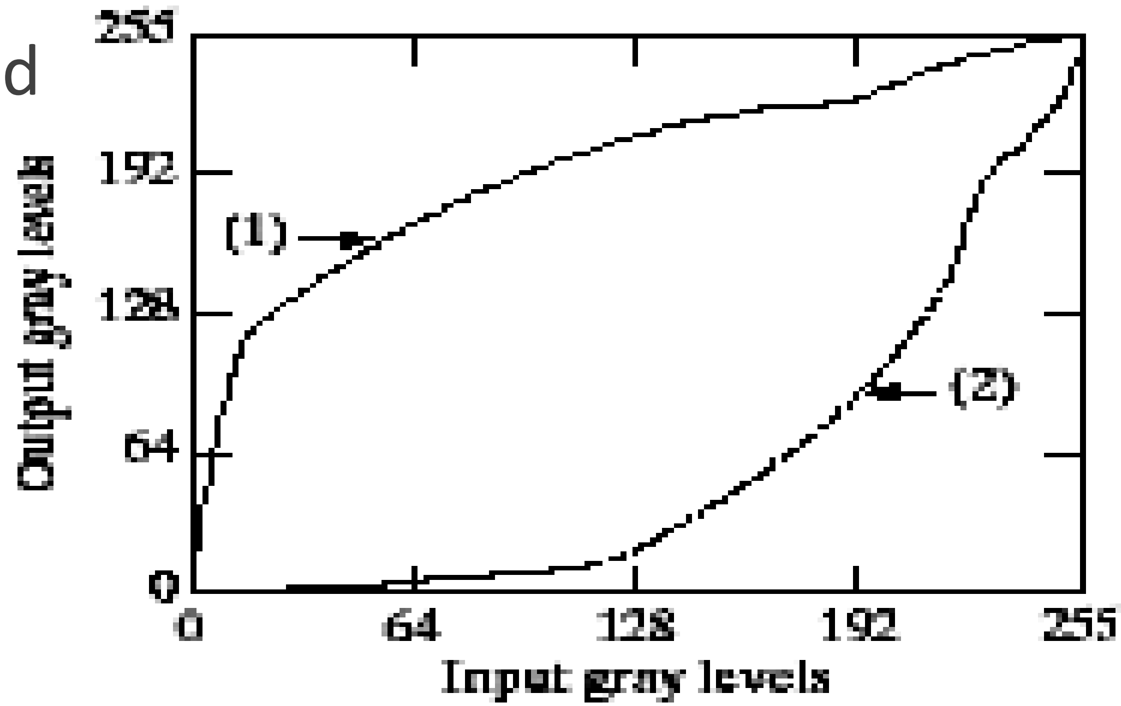
Histogram Specification

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

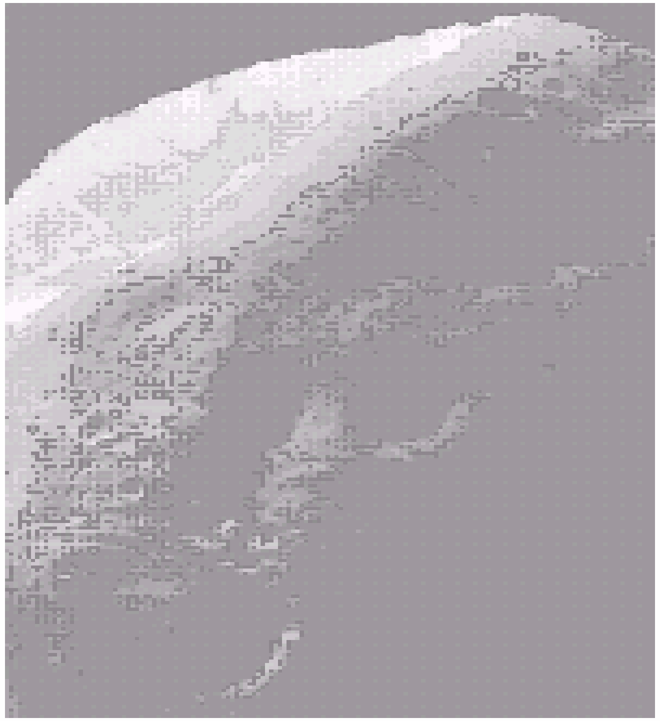
$$G(z_k) = (L-1)\sum_{i=0}^{k} p_z(z_i) = s_k$$

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

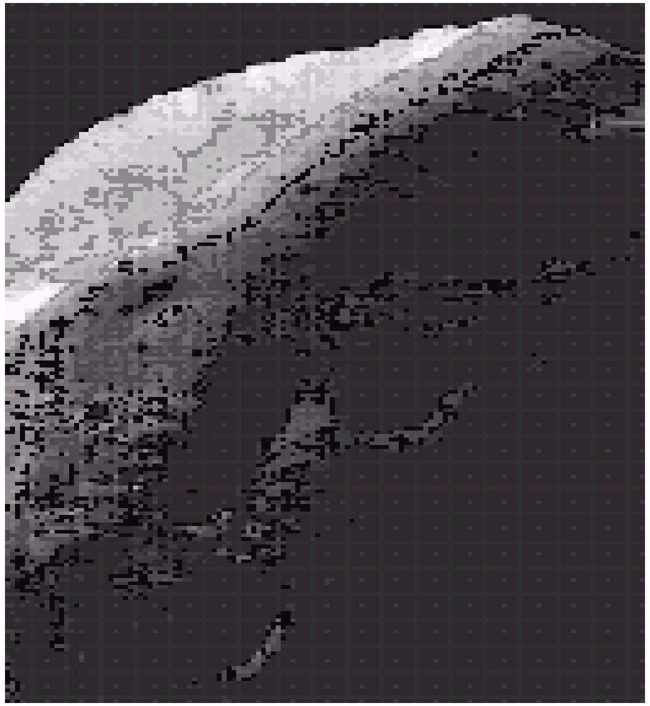
(2) the inverse transformation $G^{-1}(s)$



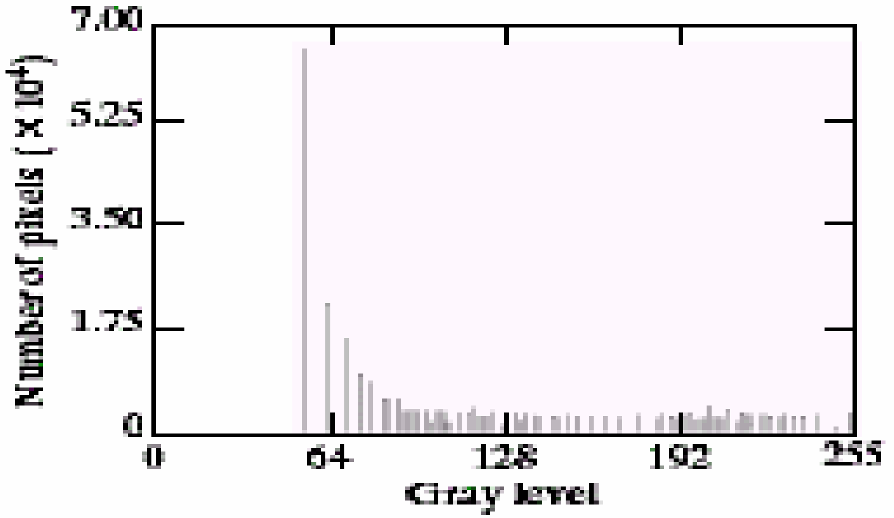
Resultant image and its histogram



Original image



Enhanced image obtained using mapping from $G^{-1}(s)$



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.

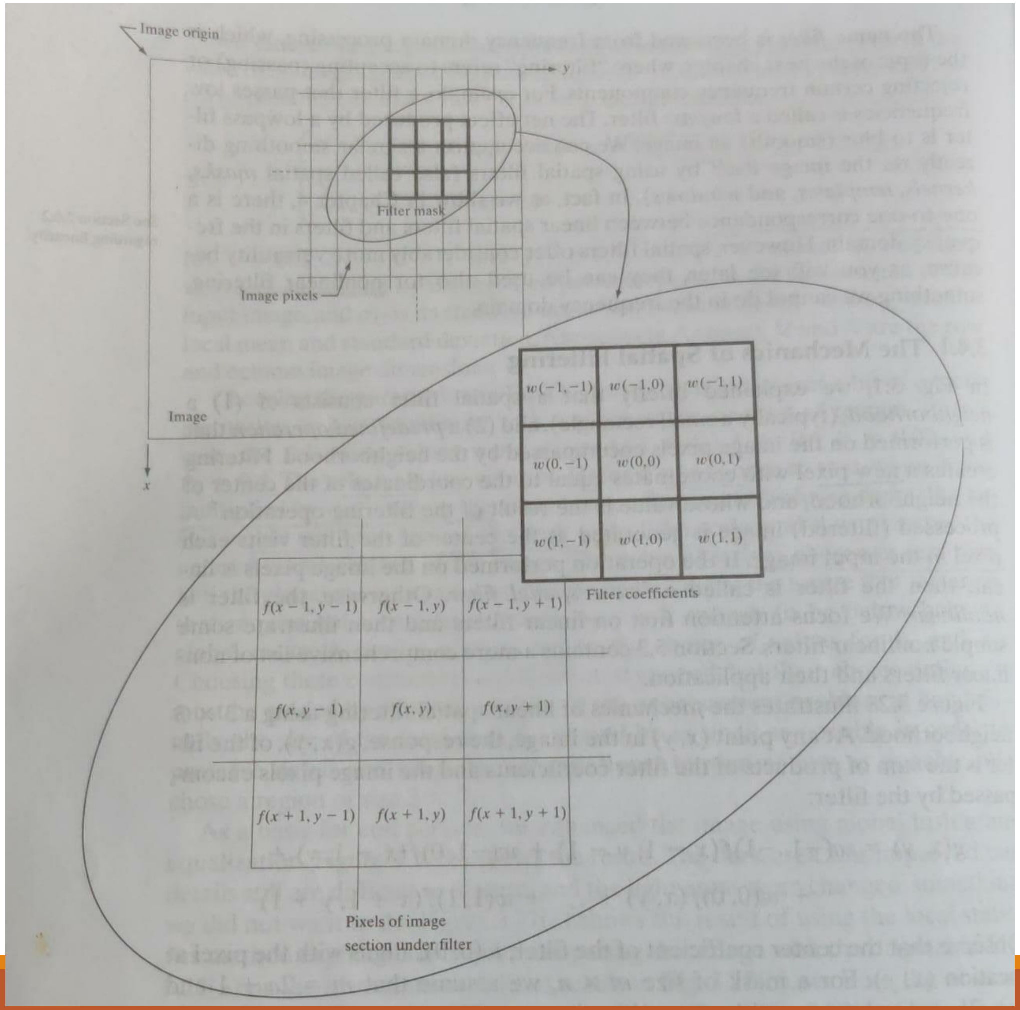
Sometimes, we may need to enhance details over small areas in an image, which is called a local enhancement.

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

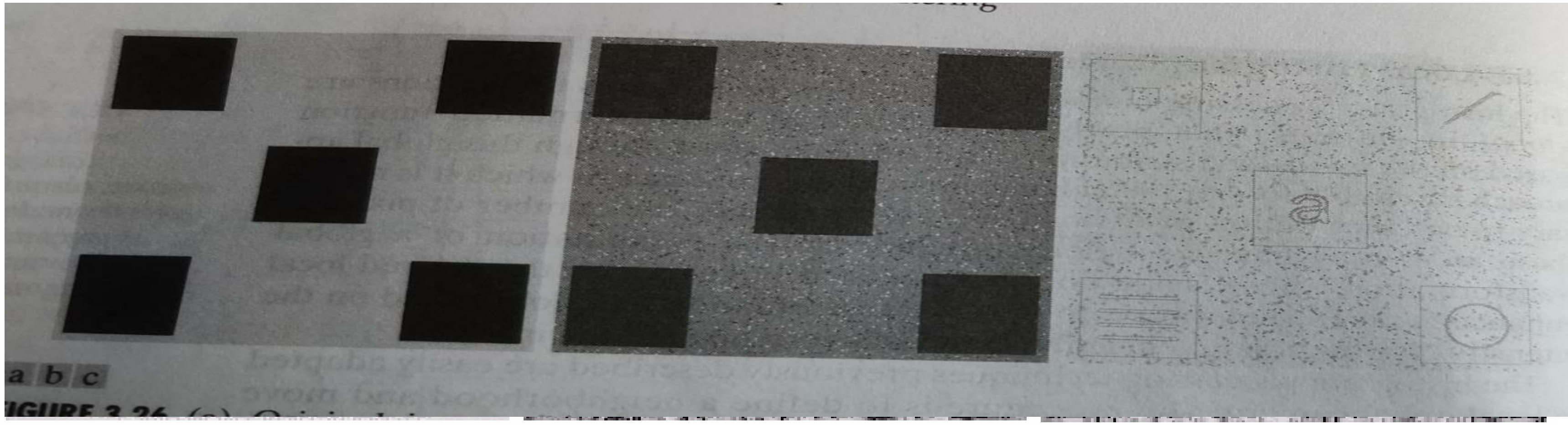


Define a square or rectangular neighborhood and move the center of this area from pixel to pixel.

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.

Local Enhancement



(a) (b) (c)

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 3x3 neighborhood (reveals the small squares inside larger ones of the original image.

Explanation of the result in (c)

- Basically, the original image consists of objects inside the larger dark ones.
- Figure (c) is obtained using local histogram equalization with a neighborhood of size 3 * 3. Here we can see significant detail contained within the dark squares.
- The intensity of these objects were too close to the intensity of the large squares, and their size were too small, to influence global histogram equalization significantly enough to show this detail.
- 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

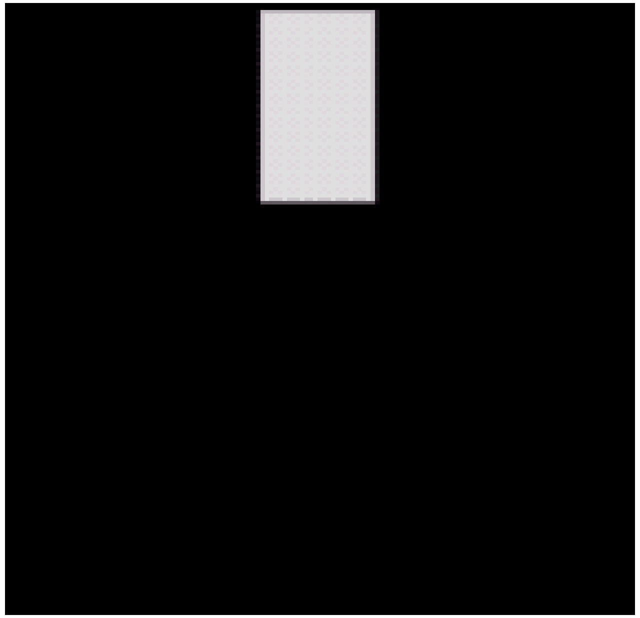
Logic operation performs on gray-level images, the pixel values are processed as binary numbers

light represents a binary 1, and dark represents a binary 0

NOT operation = negative transformation

Example of AND Operation

original image

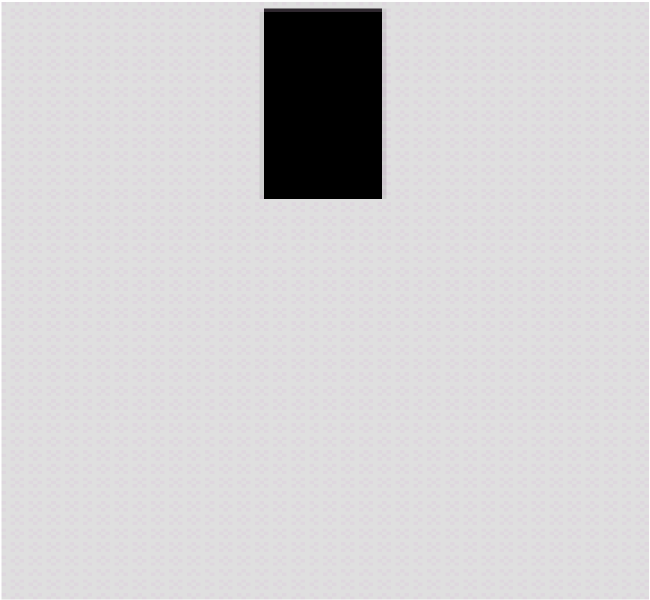


AND image mask

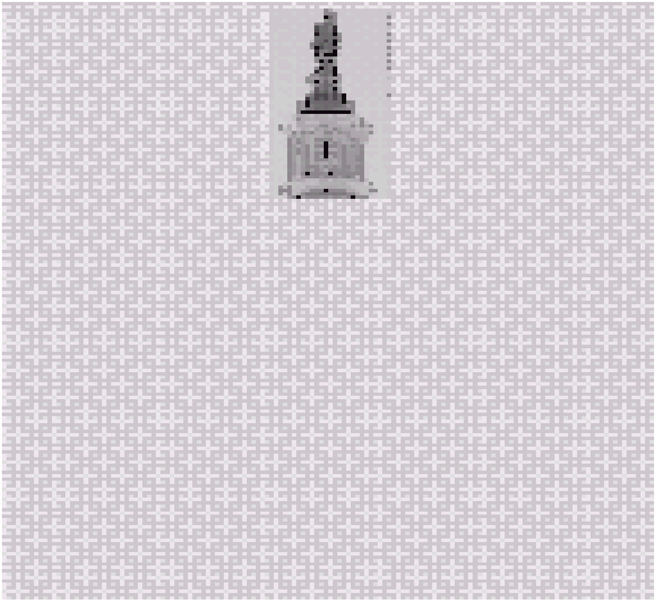
result of AND operation

Example of OR Operation

original image



OR image mask



result of OR operation

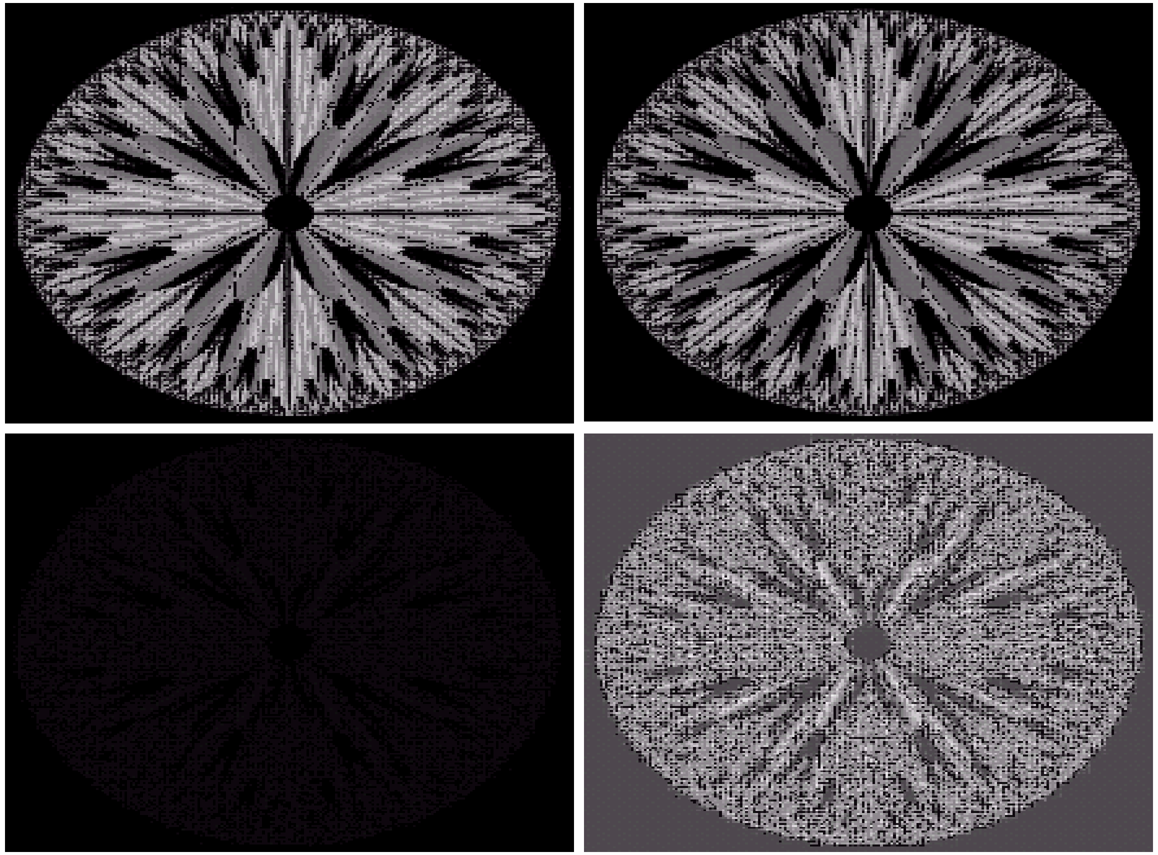
Image Subtraction

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

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)

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

Suggested Readings

- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you