Lecture 5:

Image Enhancement in Spatial Domain

Histogram equalization and matching

Local enhancement techniques

Histogram equalization

- Idea: To find a non-linear transformation

$$s = T(r)$$

  to be applied to each pixel of the input image $f(x,y)$, such that a uniform distribution of gray levels in the entire range results for the output image $g(x,y)$.

- Assuming ideal, continuous case, with normalized histograms
  - *that* $0 \le r \le 1$ *and* $0 \le s \le 1$
  - *$T(r)$ is single valued i.e., there exists* $r = T^{-1}(r)$
  - *$T(r)$ is monotonically increasing*

Histogram equalization

- $F_r(r)$ and $F_s(s)$: cdfs of original and transformed gray levels r and s.
- $p_r(r)$ and $p_s(s)$: pdfs of original and transformed gray levels r and s.

For strictly monotonically increasing transformation function

$$F_s(s) = F_r(r) \qquad \text{or} \qquad p_s(s)\, ds = p_r(r)\, dr$$

- Goal of histogram equalization:

  Gray levels are uniformly distributed

  i.e. pdf $p_s(s) = 1$ over the range $0 \le s \le 1$

$$p_s(s) = p_r(r)\left(\frac{dr}{ds}\right) = 1 \qquad \text{or} \qquad p_r(r) = \frac{ds}{dr} = \frac{dT(r)}{dr}$$

$$\boxed{\Rightarrow s = T(r) = \int_0^r p_r(\omega)\, d\omega}$$

Histogram equalization

If the following transformation function is used

$$s = T(r) = \int_0^r p_r(\omega)d\omega \quad \text{for} \qquad 0 \le r \le 1$$

Then the pdf $p_s(s) = 1$ over the range $0 \le s \le 1$

In words

If we select $T(r)$ as the cumulative distribution of r

Then the output image will have a uniform pdf of gray levels

Now Consider

1. a digital (gray level) case
2. the gray levels $0 \le r \le L-1$

Histogram equalization

The discrete approximation of the transformation function for histogram equalization is:

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

where $p_r(r_j) = \dfrac{n_j}{n}, \quad j = 0, \ldots\ldots\ldots, L-1 \quad \textit{and} \quad n = \sum_{j=0}^{L-1} n_j$

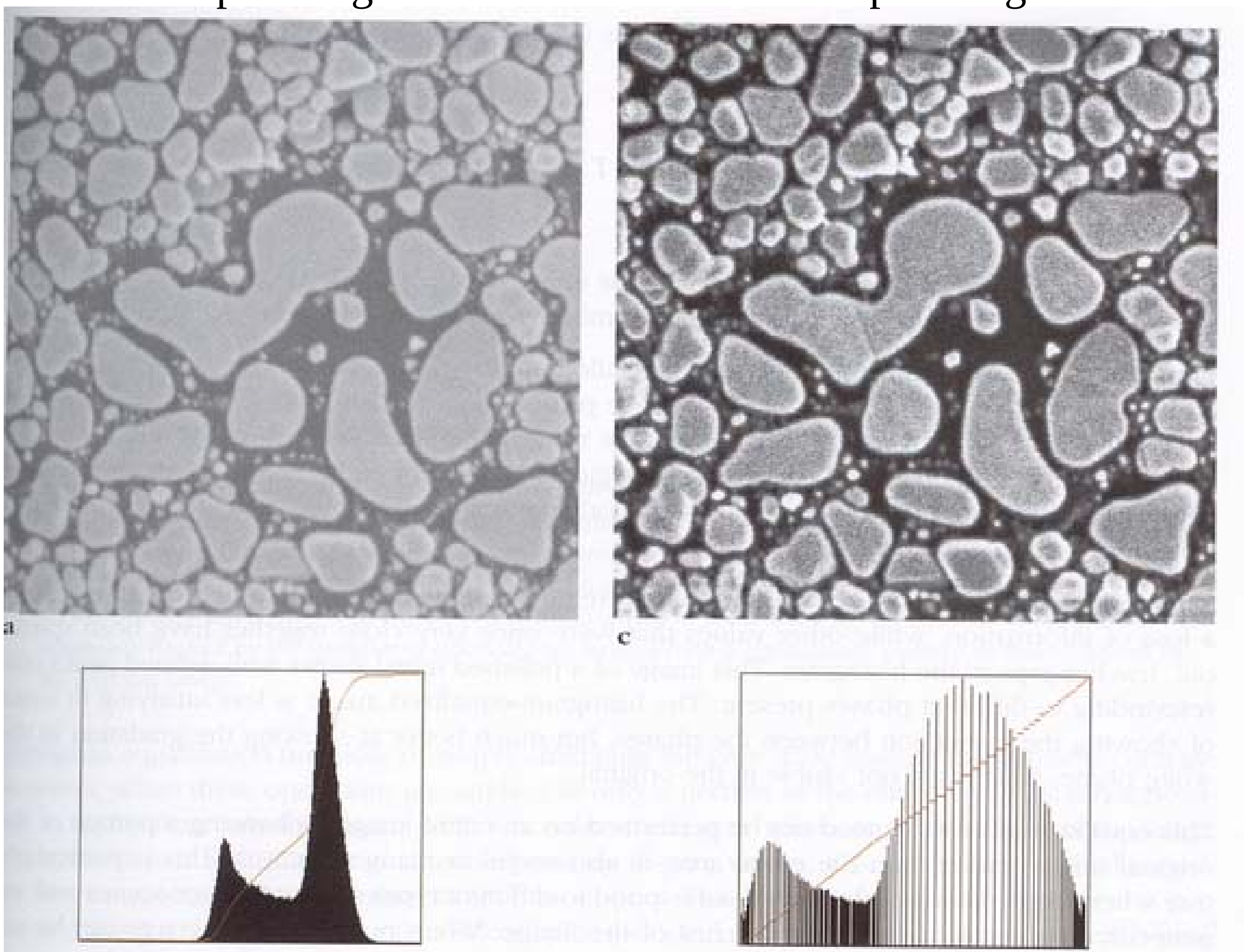
$\boldsymbol{n_j}$: number of pixels with gray level $\boldsymbol{r_j}$

$\boldsymbol{n}$: total number of pixels

Note: For digital images, gray-level pdf cannot be exactly uniform after histogram equalization

Histogram equalization examples

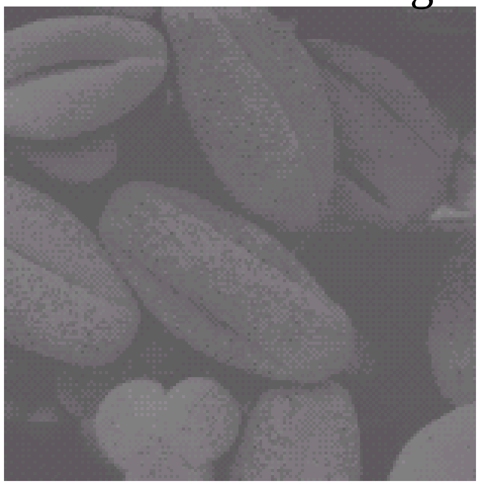
Input image Output image



Input histogram and cdf Output histogram and cdf

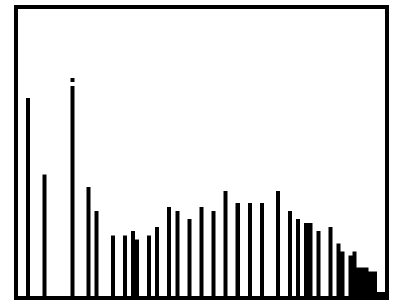
Histogram equalization examples

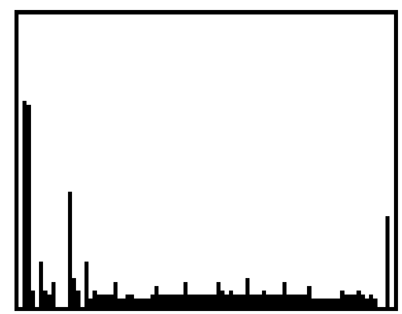
Low contrast image



Output image

Equalized histogram





high contrast image

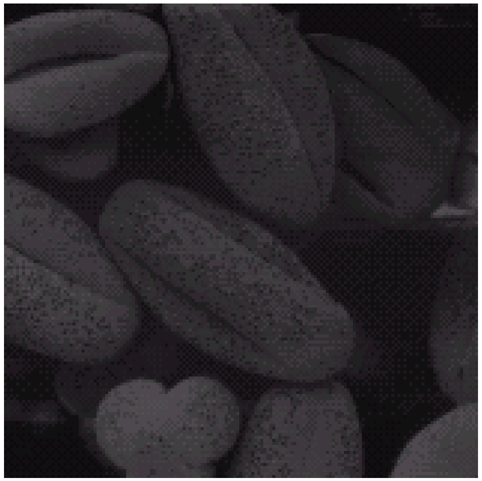
Output image

Equalized histogram

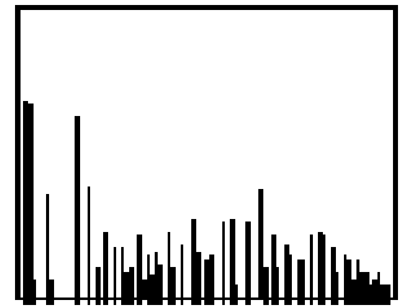
Histogram equalization examples

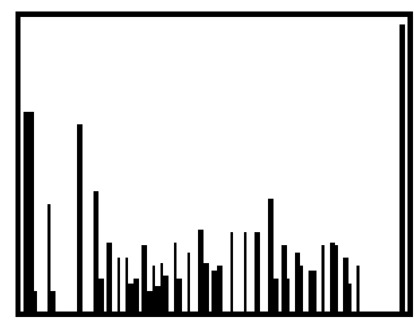
Dark input image

Output image



Equalized histogram



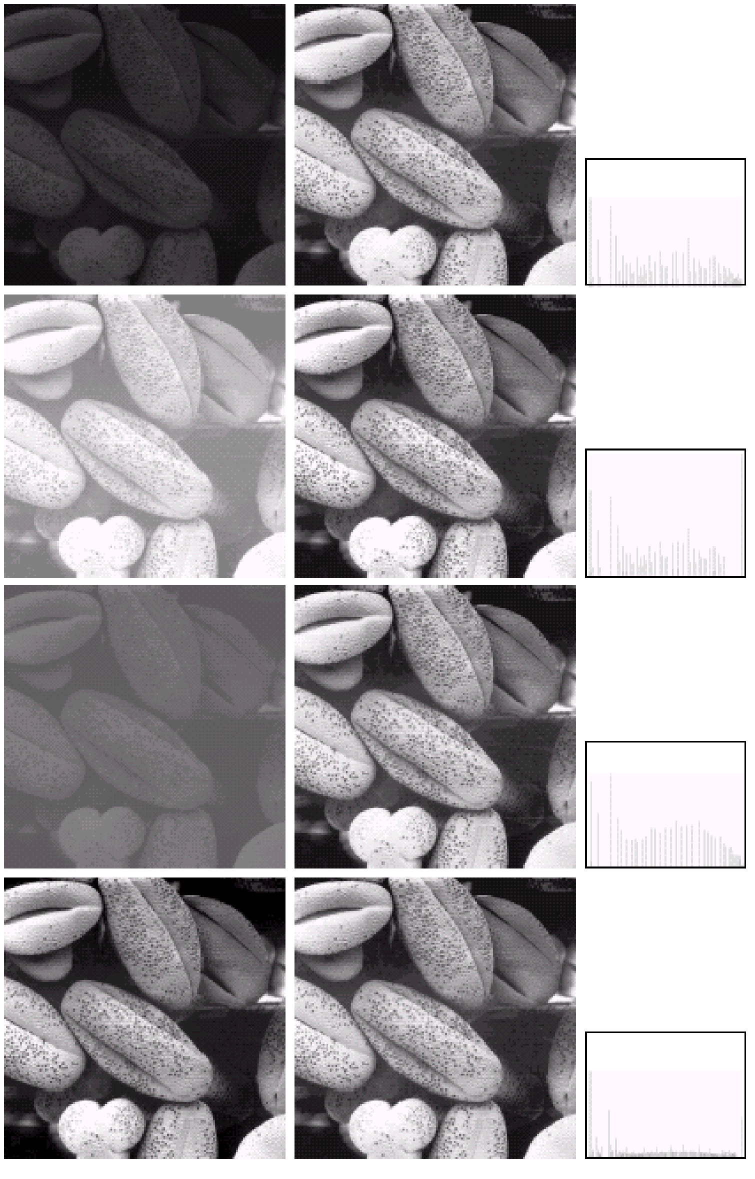


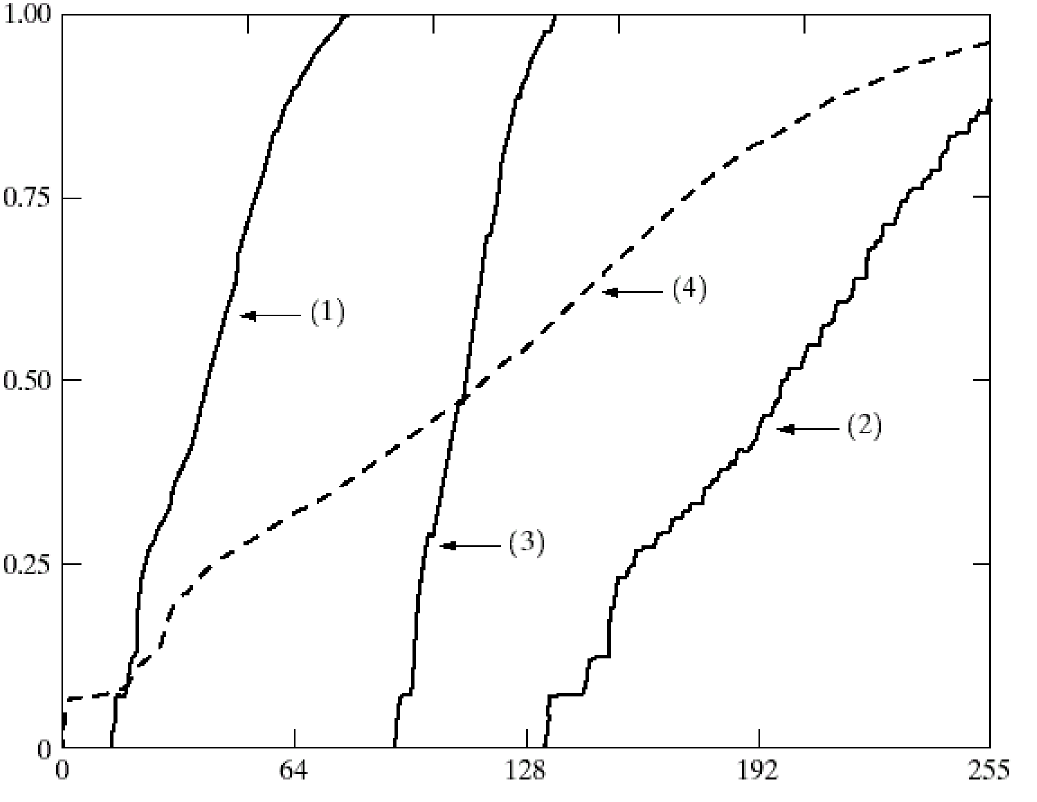
Equalized histogram

Bright input image

Output image

Histogram equalization examples





Transformation functions for histogram equalization

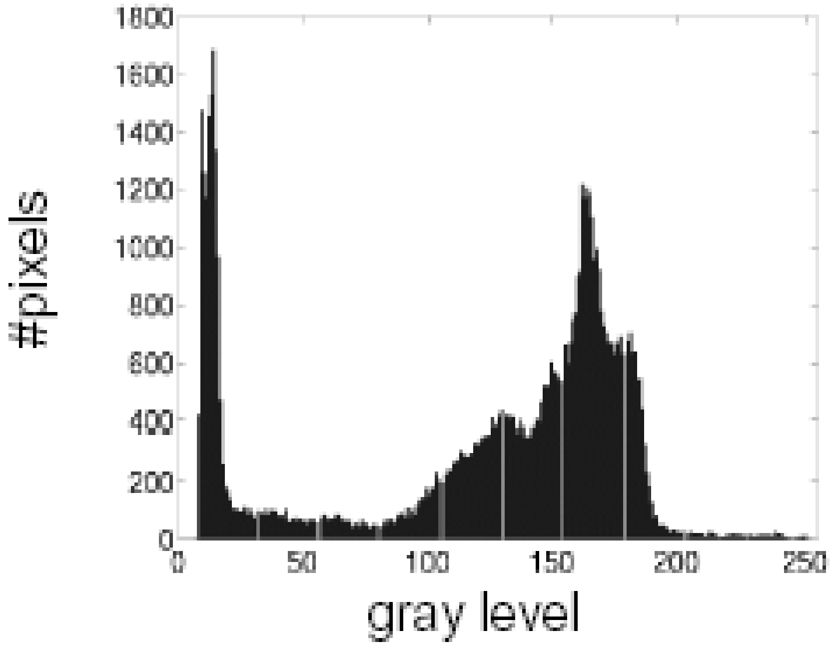
Histogram equalization examples

Original image
Cameraman

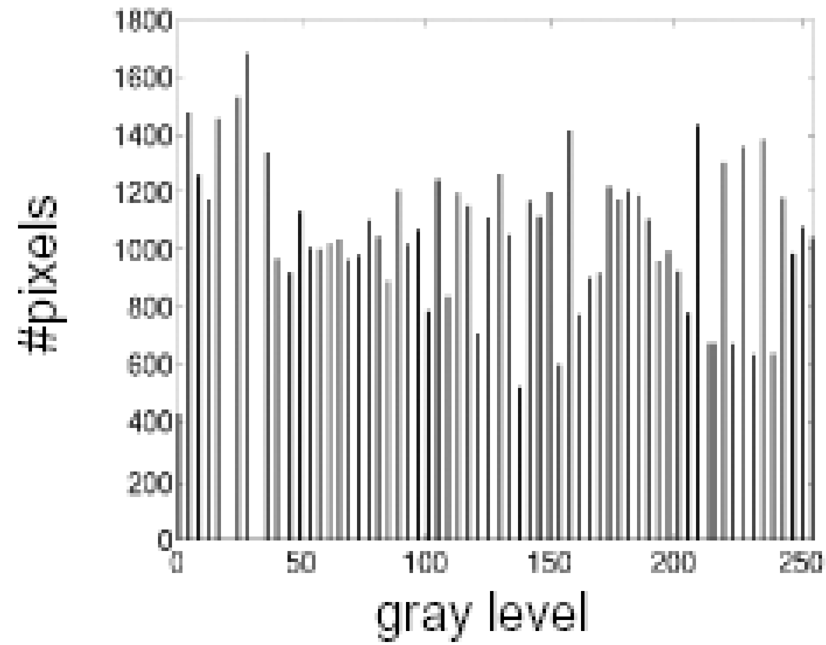
Cameraman
after histogram equalization

Histogram equalization examples

Original image *Cameraman*



. . . after histogram equalization



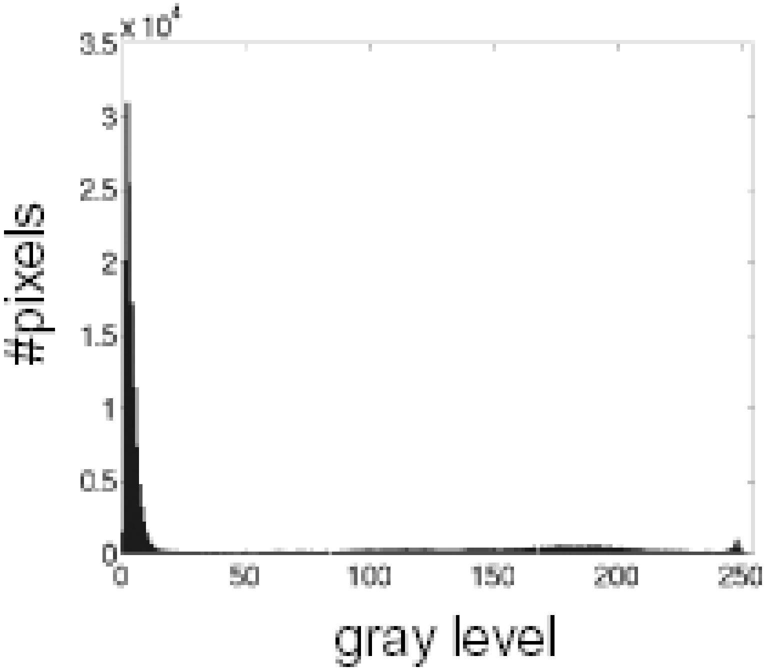
Histogram equalization examples

Original image *Moon*

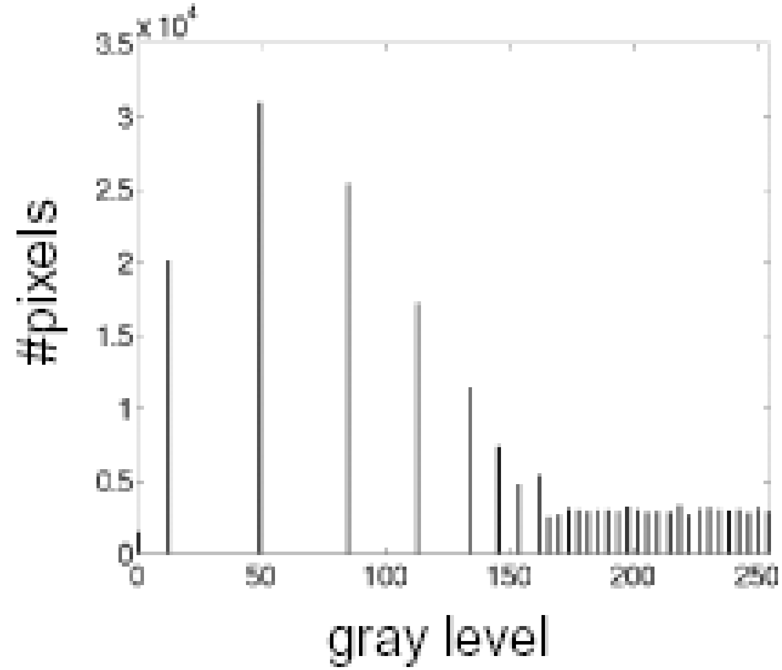
Moon
after histogram equalization

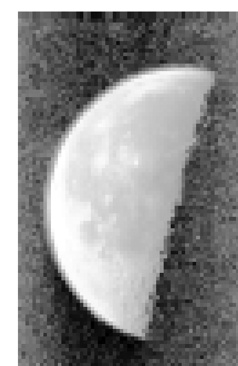
Histogram equalization examples

Original image *Cameraman*



. . . after histogram equalization





Histogram specification/matching

Histogram equalization method:

- Only generates one result: an image with approximately uniform histogram (without any flexibility)
- Enhancement may not be achieved as desired

Histogram specification:

Transform an image according to a specified gray-level histogram

Includes

- Specify particular histogram shapes ($p_z(z)$) capable of highlighting certain gray-level ranges
- Obtain the transformation function for transformation of r to z

Histogram specification/matching

Method (continuous case)

Notation:

r : gray level of input image (pdf: $pr(r)$)

z : gray level of desired output image (pdf: $pz(z)$)

$$r \rightarrow s = T(r) = \int_0^r p_r(\omega) d\omega$$

$$z \rightarrow v = G(z) = \int_0^z p_z(t) dt$$

s & v represent gray levels of histogram-equalized images hence $s \approx z$

To obtain the transformed gray levels, we can apply:

$$z = G^{-1}(s) = G^{-1}[T(r)]$$

Histogram specification/matching

Procedure of applying histogram specification

- Obtain transformation function $T(r)$: integral of $p_r(r)$
- Obtain transformation function $G(z)$: integral of $p_z(z)$
- Obtain the inverse function $G^{-1}(\cdot)$
- Finally, output image: $z = G^{-1}(T(r))$

Modification for discrete case

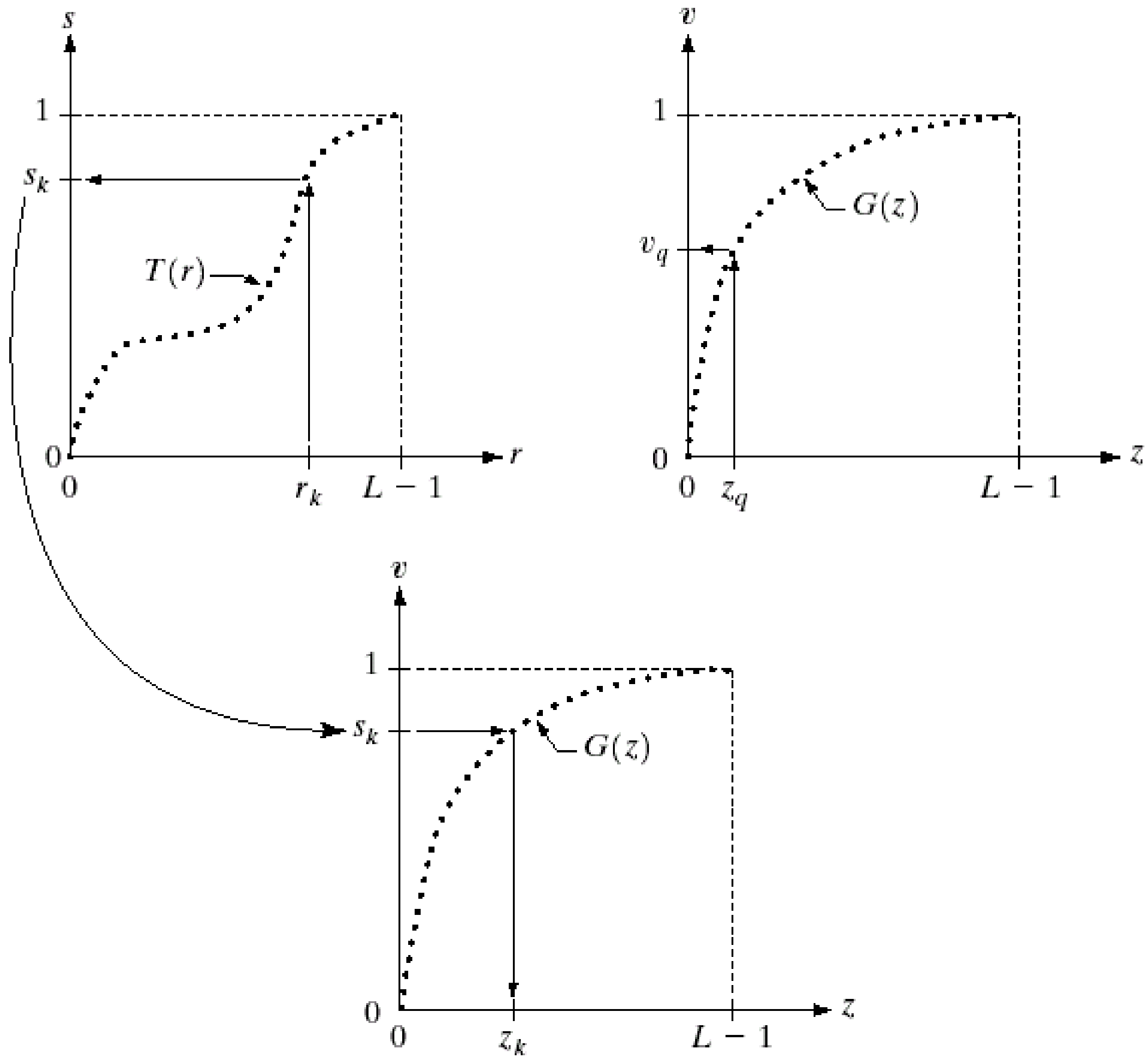
$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}, \quad k = 0,1,2,.....,L-1.$$

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

$$z_k = G^{-1}(s_k) = G^{-1}[T(r_k)], \quad k = 0,1,2,.....,L-1.$$

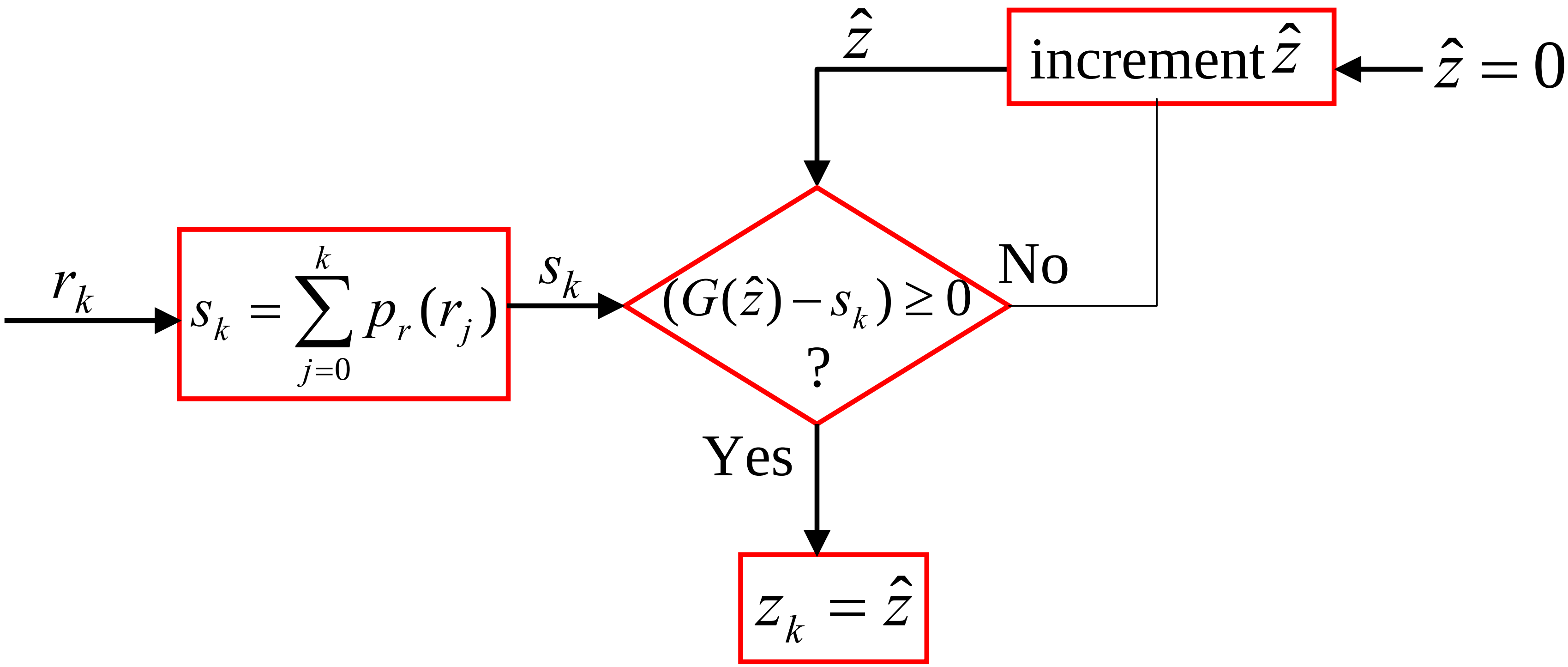
Histogram specification/matching

(a) Graphical interpretation of mapping from r_k to s_k via $T(r)$. (b) Mapping of z_q to its corresponding value v_q via $G(z)$. (c) Inverse mapping from s_k to its corresponding value of z_k.



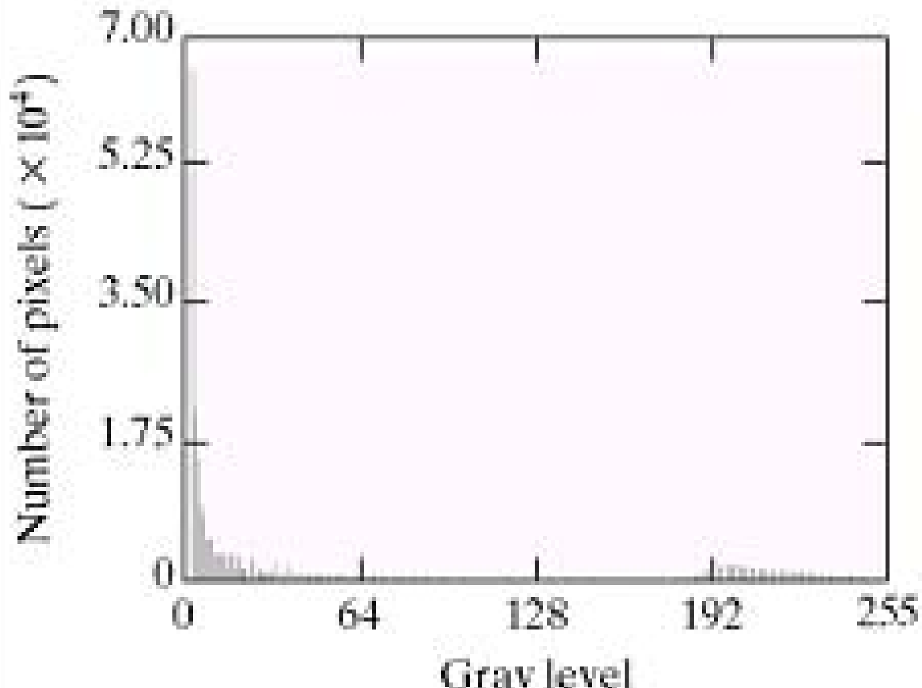
Histogram specification/matching

Analytical expressions for $G^{-1}(T(r_k))$ are difficult, but for discrete case it can be mapped in following way



Histogram matching example

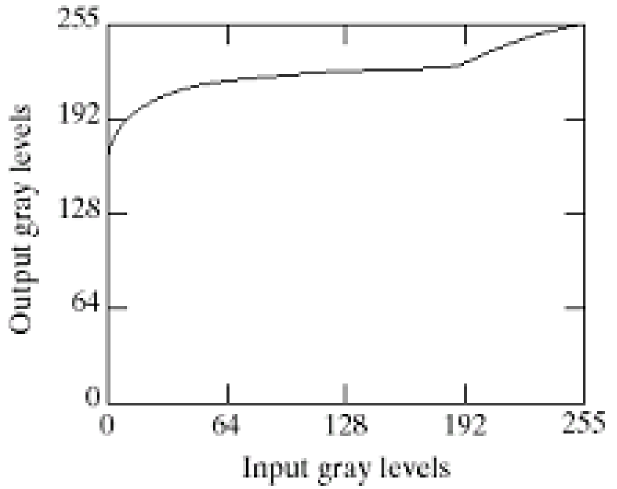
Image of Mars



Histogram

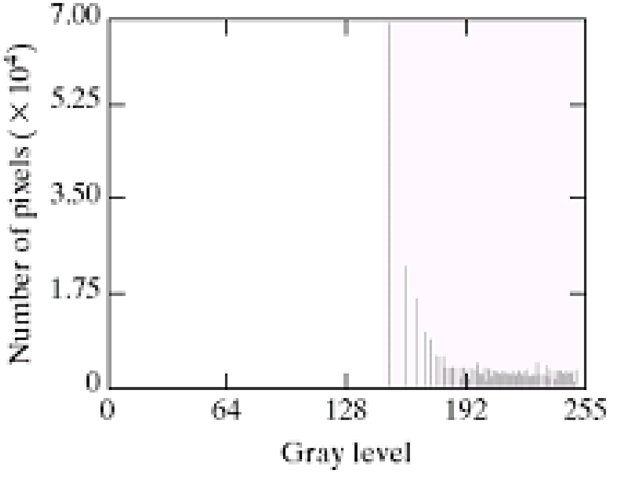
Histogram matching example(continued)

Transformation Function



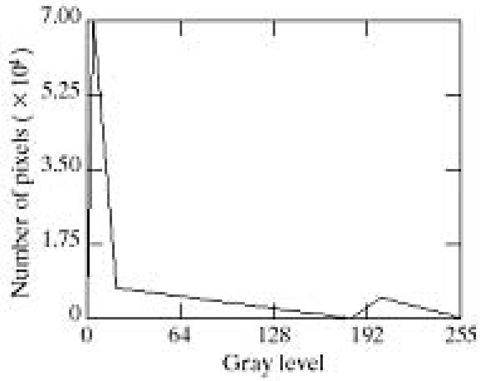
Histogram Equalized Image

Histogram of New Image



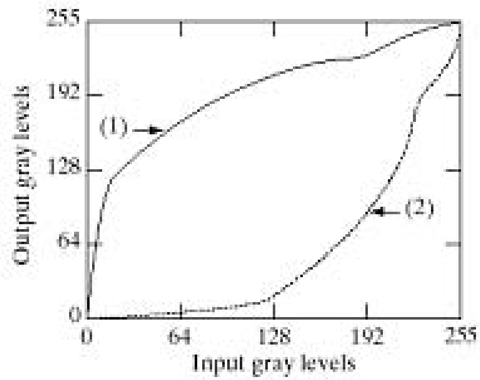
Histogram matching example(continued)

Specified Histogram

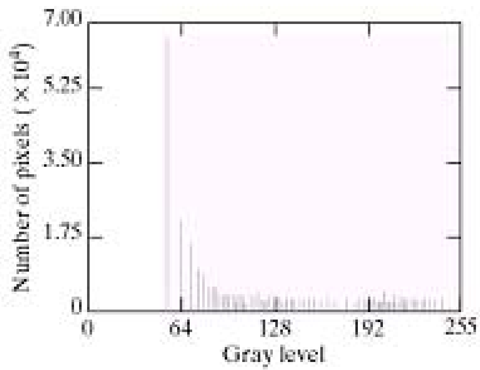


Enhanced Image using Mappings from Curve (2)

Transfer function and inverse transfer function



Histogram of Image



Local histogram processing

The histogram processing methods mentioned up to now are global transformation where:

Function is designed according to the gray-level distribution over an entire image

Global transformation methods may not be suitable *for enhancing details over small areas*

(where number of pixels may have negligible influence on designing the global transformation function)

Local histogram processing

Implementation steps

1: Define a square or rectangular neighborhood (block), compute the histogram in the local block

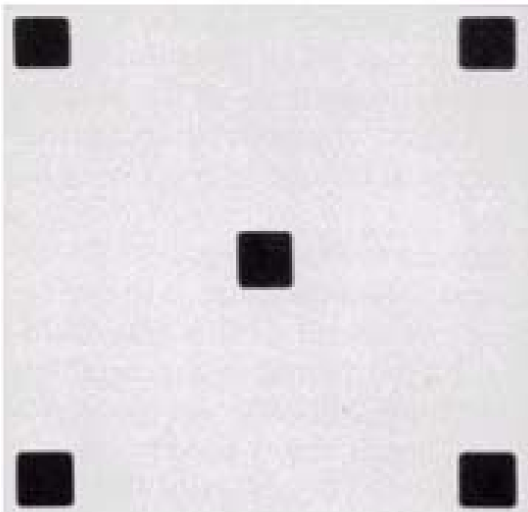
2: Utilize the histogram equalization or specification method to generate the transformation function, perform the gray level mapping for each pixel in the block

3: Move the center of the block to an adjacent pixel location and repeat the procedure

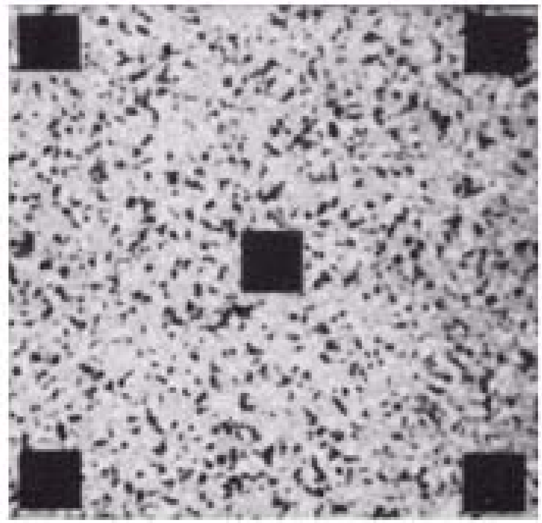
Note

- The local histogram can be updated each time without re-computing the histogram over all pixels in the new block (since the block only shifts one pixel each time)
- If utilizing non-overlapping region shift, the processed image usually has an undesirable checkerboard effect

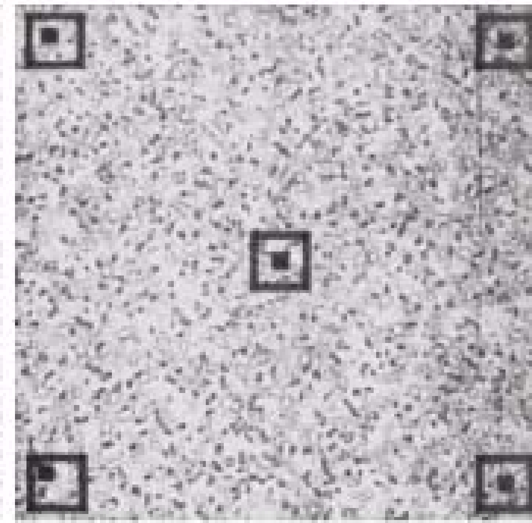
Local histogram equalization example



Original Image

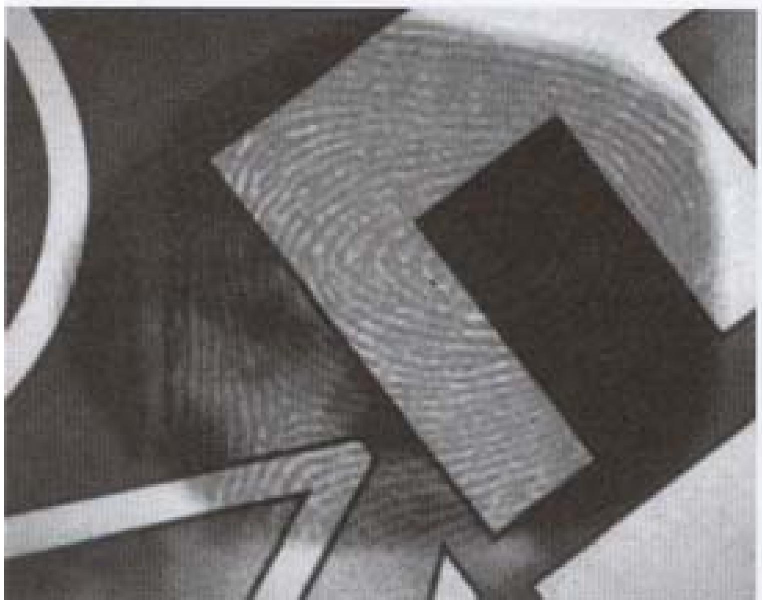


Result of Global Histogram Equalization



Result of local Histogram Equalization

Local histogram equalization example



Original Image

Result of local Histogram Equalization using 9x9 window

Local enhancement using statistical parameters from histogram

Local enhancement can be based on statistical properties of the gray levels in a block instead of using full histogram

Examples:

- Mean gives the average brightness of the image
- Variance (σ^2) and its square root the standard deviation gives the deviation of intensities on average from the mean value (average contrast)

Local enhancement using statistical parameters

S_{xy} : a neighborhood (subimage) of size N_{Sxy}; a block centered at (x,y)

$m_{S_{xy}}$: gray-level mean in S_{xy}
$$m_{S_{xy}} = \frac{1}{N_{S_{xy}}} \sum_{(s,t)\in S_{xy}} f(s,t)$$

$\sigma^2_{S_{xy}}$: gray-level variance in S_{xy}
$$\sigma^2_{S_{xy}} = \frac{1}{N_{S_{xy}}} \sum_{(s,t)\in S_{xy}} \left(f(s,t) - m_{S_{xy}}\right)^2$$

$\sigma_{S_{xy}}$: standard deviation, square root of variance $\sigma^2_{S_{xy}}$

M_G : global mean of $f(x,y)$

D_G : global standard deviation of $f(x,y)$

Local enhancement using statistical parameters

The statistical parameters can be used in various ways:

- For the direct calculation of transformation function (adaptive transformation function) for example

$$g(x,y) = A_{S_{xy}}[f(x,y) - m_{S_{xy}}] + m_{S_{xy}},$$

where $A_{S_{xy}}$ is the local gain factor, $A_{S_{xy}} = \dfrac{kM_G}{\sigma_{S_{xy}}}, \quad 0 < k < 1$

- Using them in defining ranges for different transfer functions for example

$$g(x,y) = \left\{ \begin{array}{ll} E.f(x,y) & \text{if } m_{S_{xy}} \le k_0 M_G \text{ AND } k_1 D_G \le \sigma_{S_{xy}} \le k_2 D_G \\ f(x,y) & \text{otherwise} \end{array} \right\}$$

where E, k_0, k_1, k_2, are specified parameters

Local enhancement with adaptive transformation function

Original moon image

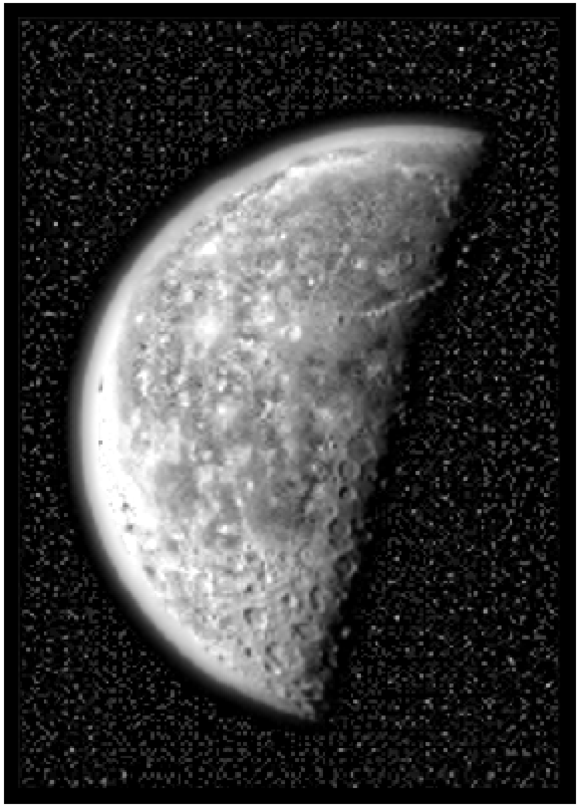


Image enhanced using adaptive transformation, window size: 15x15, k = 0.5

Histogram equalized image

Local enhancement using statistical parameters

Defining ranges for different transformation functions

$$g(x,y)=\left\{\begin{array}{ll} E.f(x,y) & \text{if } m_{S_{xy}} \le k_0 M_G \text{ AND } k_1 D_D \le \sigma_{S_{xy}} \le k_2 D_G \\ f(x,y) & \text{otherwise} \end{array}\right\}$$

FIGURE 3.24 SEM image of a tungsten filament and support, magnified approximately 130×.

Local enhancement using statistical parameters

Image formed from the local means

Image formed from the local standard deviations

Image formed from the multiplication constants selected for different mean/std ranges

$E = 4.0$, $k_0 = 0.4$, $k_1 = 0.02$, and $k_2 = 0.4$,

Local enhancement using statistical parameters

Input microscopic image

Enhanced output image

Mathematical/logical operations on images

- Addition
  - *Averaging images for noise removal*
- Subtraction
  - *Removal of background from images*
  - *Image enhancement*
  - *Image matching*
  - *Moving/displaced object tracking*
- Multiplication
  - *Superimposing of texture on an image*
  - *Convolution and correlation of images*
- And and or operations
  - *To remove the unnecessary area of an image through mask operations*

Image averaging for noise reduction

A noisy image can be represented by

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

where $\eta(x,y)$ denotes the noise in the image

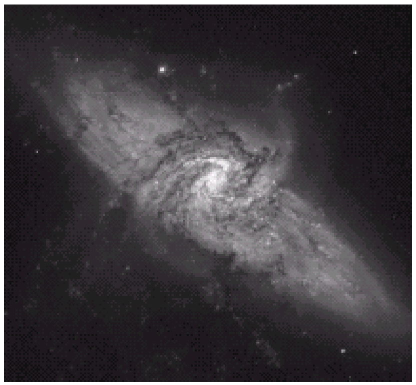
Since the noise is random and the content $f(x,y)$ is fixed,

The noise can be removed by taking more noisy images of the same object and averaging them out

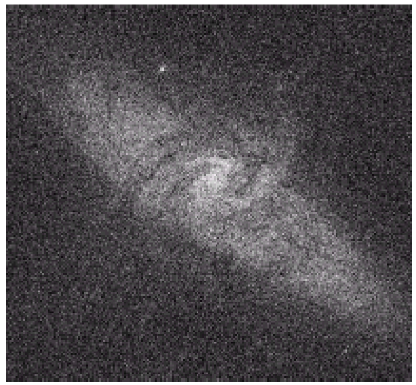
$$\overline{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y),$$

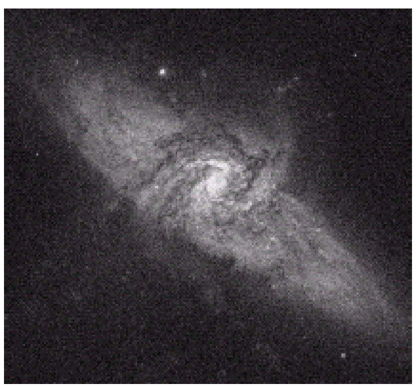
Image averaging for noise reduction

Original image

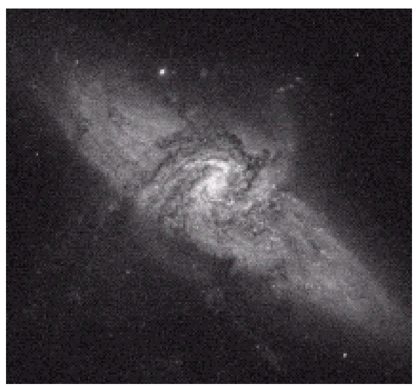


Noisy image

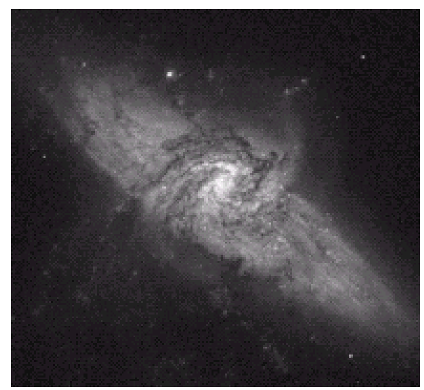




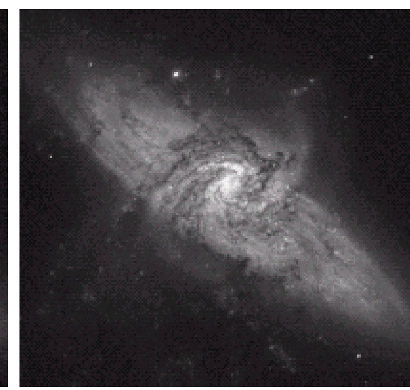
Result of averaging using 8 noise samples



Using 16 noise samples



Using 64 noise samples

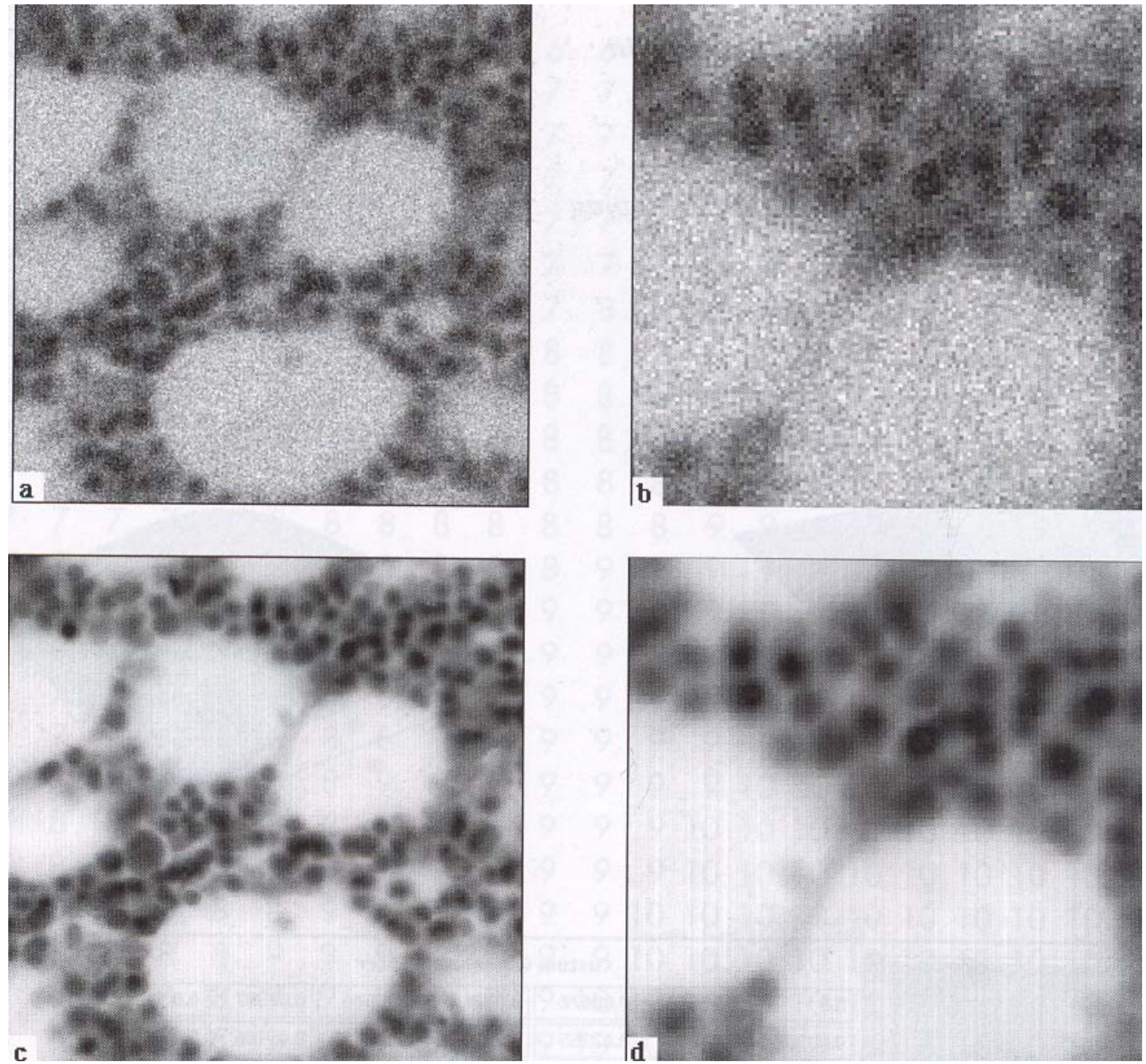


Using 128 noise samples

Image averaging for noise reduction

Noisy image

Noise reduction by averaging 256 samples



Examples of image subtraction

Original image

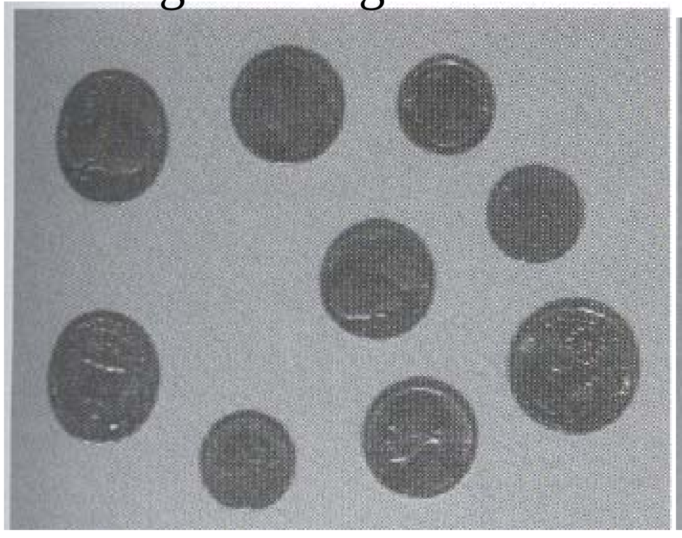
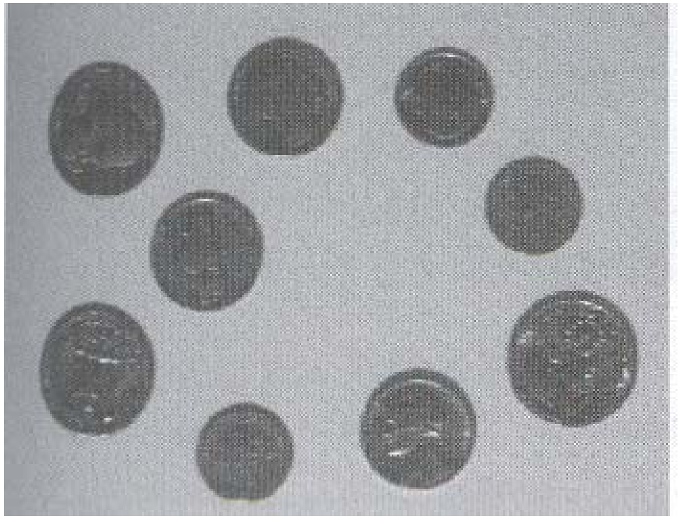
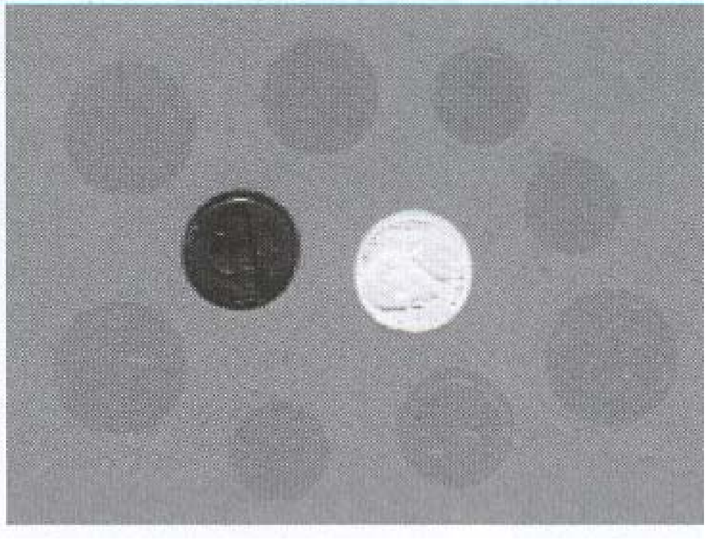


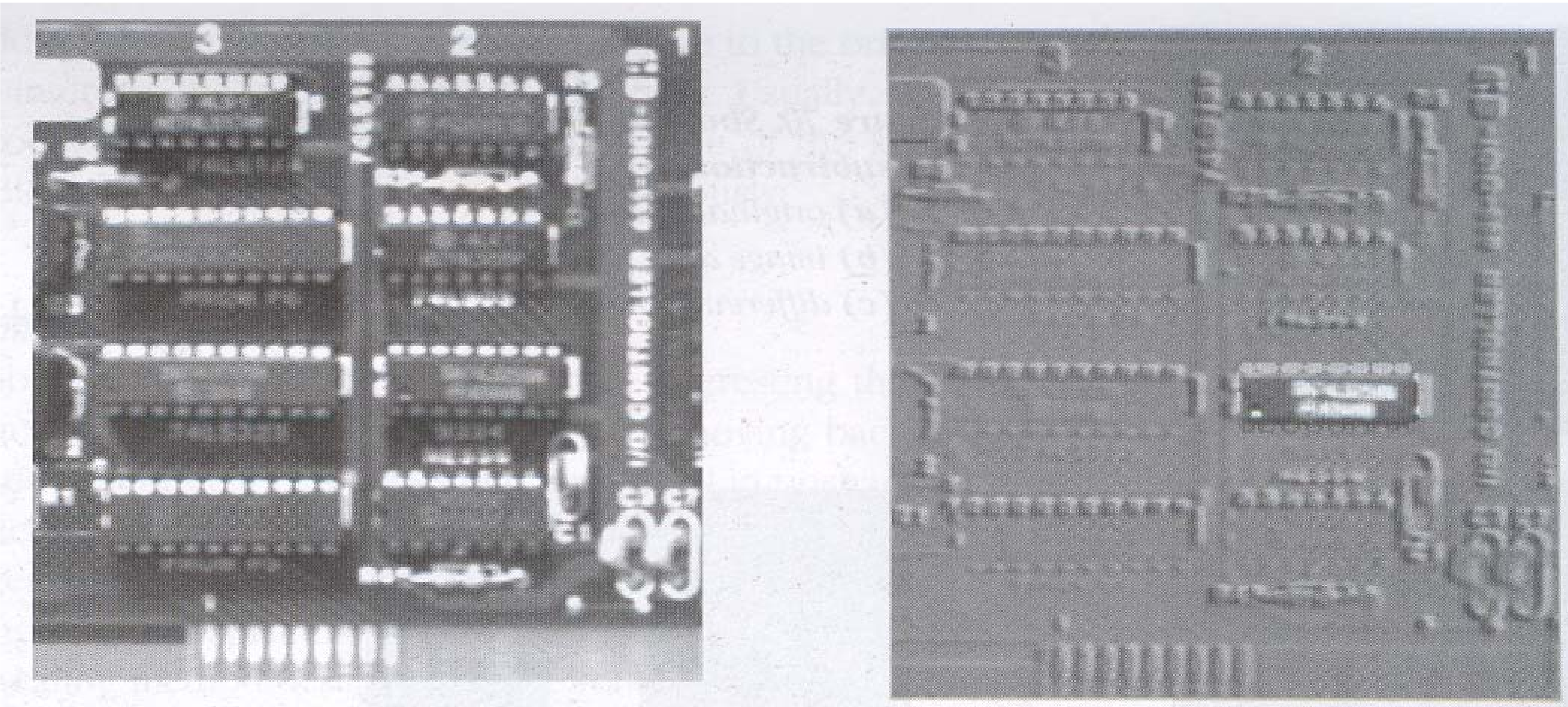
Image after moving one coin





Difference image after pixel by pixel subtraction of second image from first image

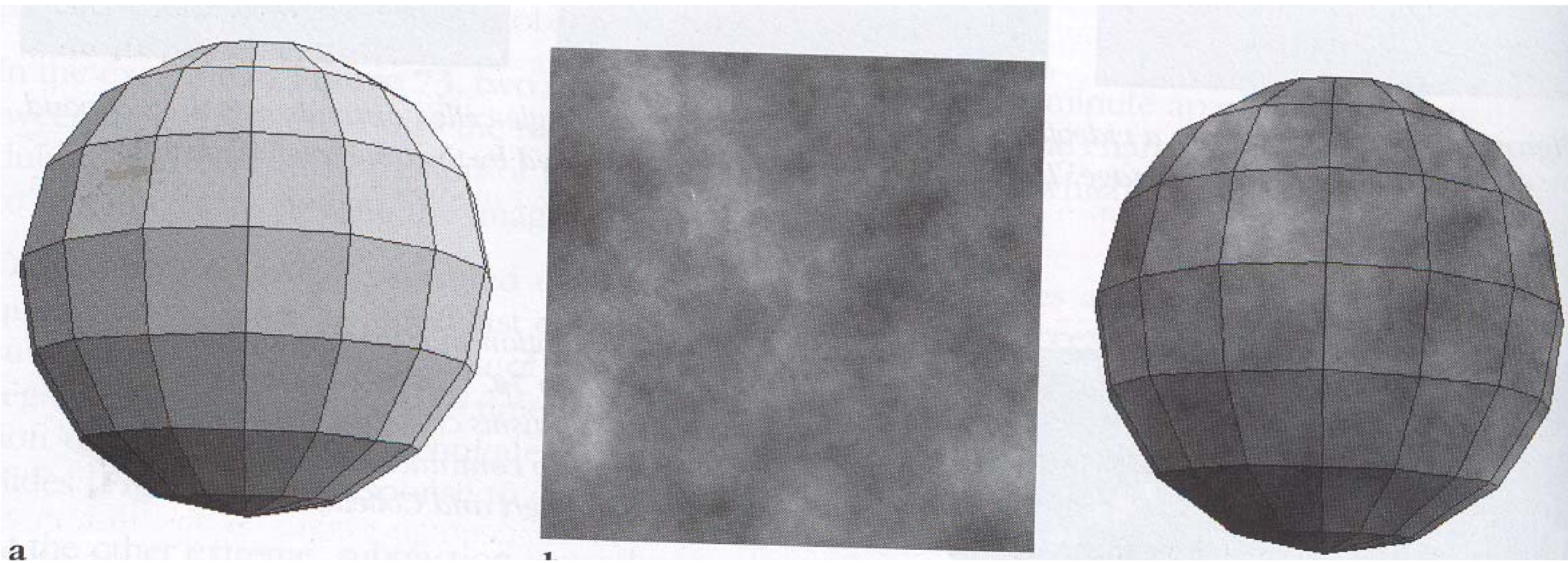
Examples of image subtraction



Difference of images from quality control: a missing chip in PCB is detected by subtracting the master image from image of each sample

Examples of image Multiplication

Multiplication of images can be used for superimposing texture on an image



Smooth spherical surface image

Texture to be superimposed

output image

Example of logical operations using masks

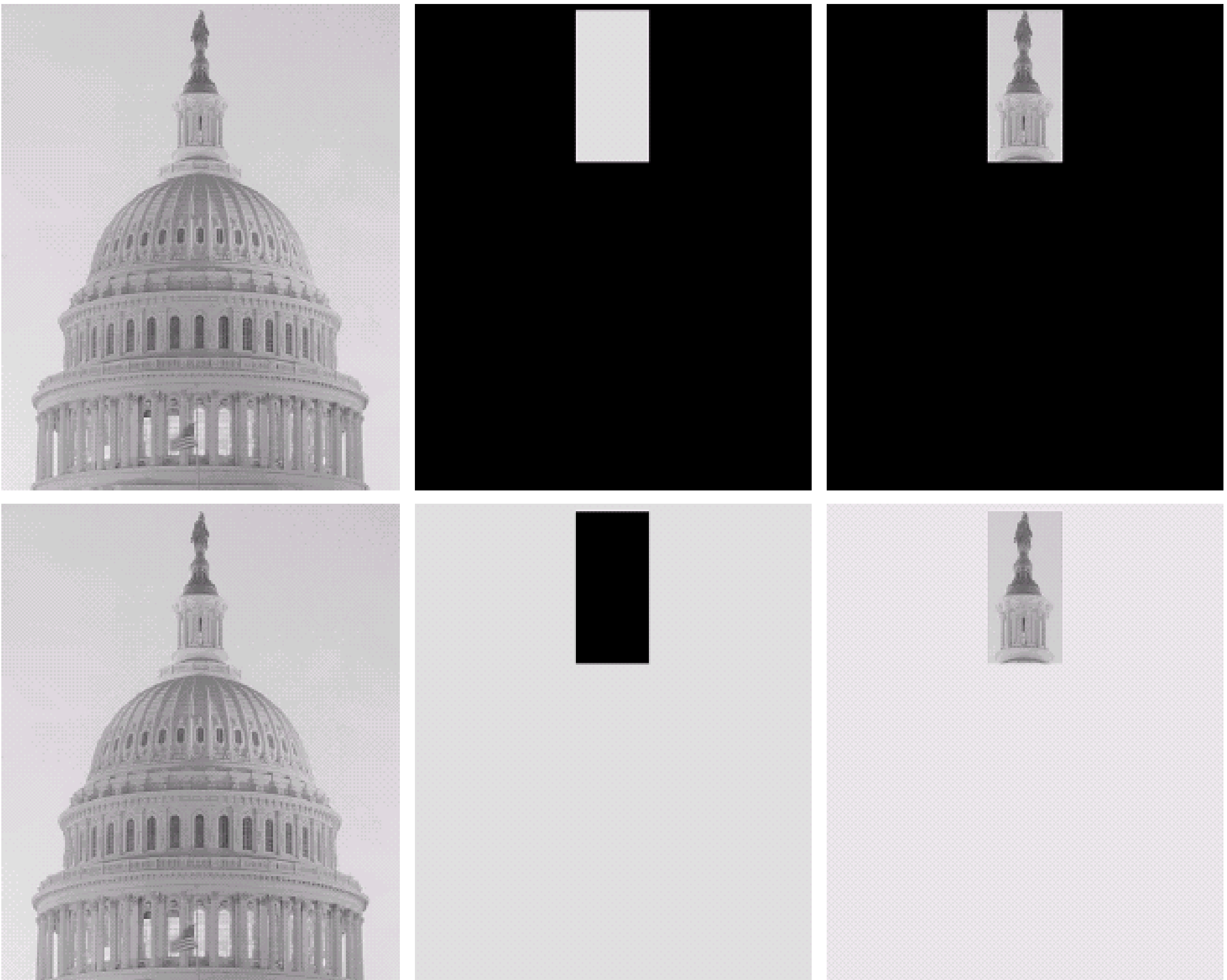


FIGURE 3.27 (a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).