Computer Vision

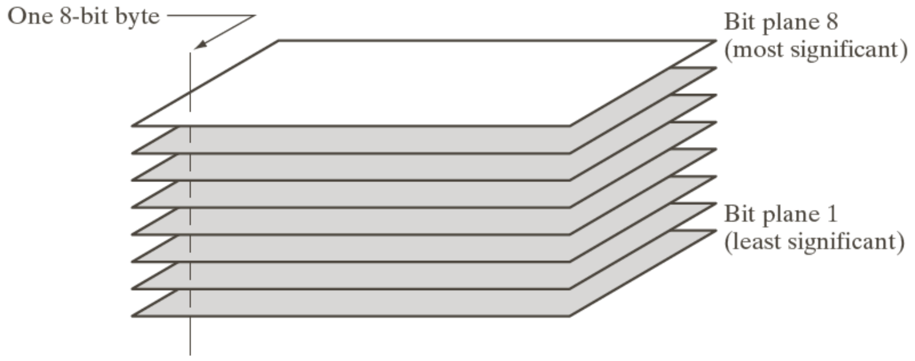
Course 4

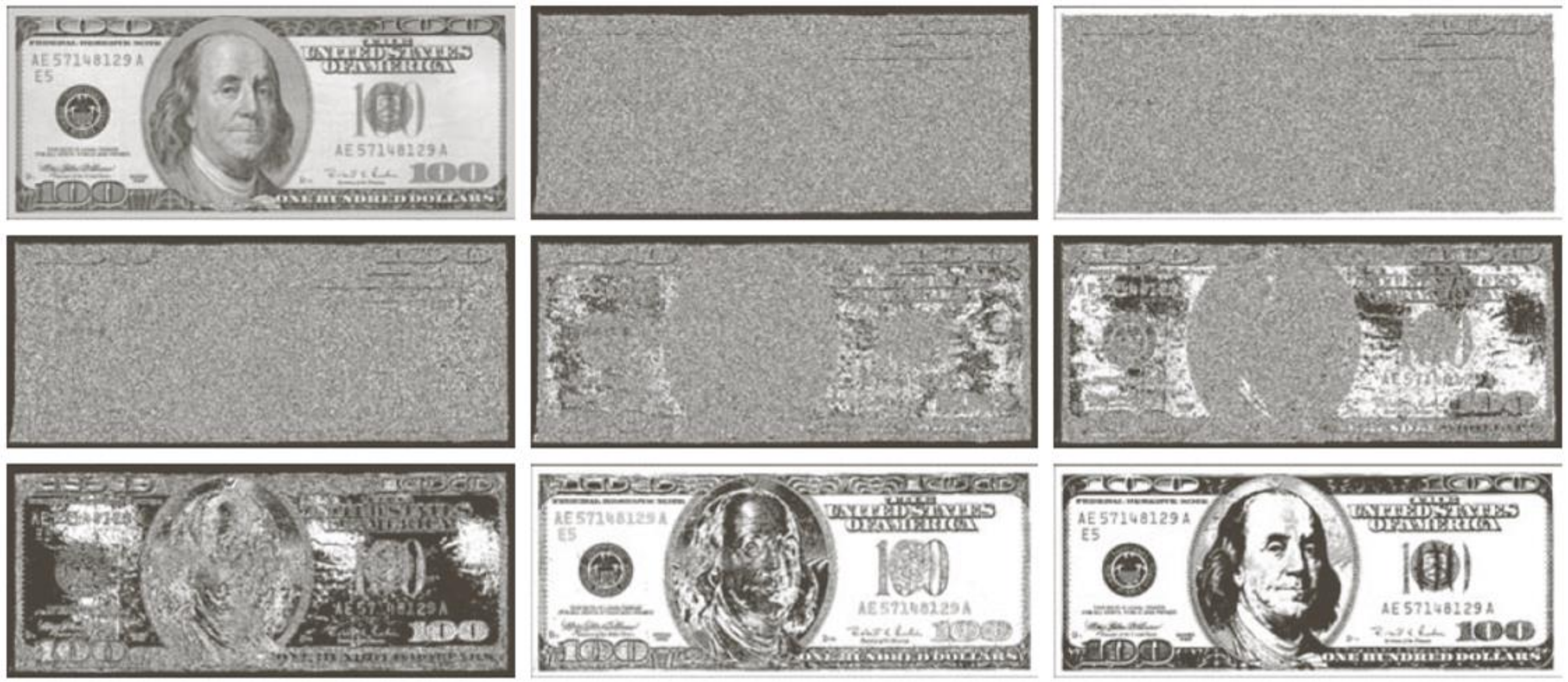
Course 4

Bit-plane slicing

For an 8-bit image, ***f(x, y)*** is a number in ***[0,255]***, with 8-bit representation in base 2

This technique highlights the contribution made to the whole image appearances by each of the bits. An 8-bit image may be considered as being composed of eight 1-bit planes (plane 1 – the lowest order bit, plane 8 – the highest order bit)





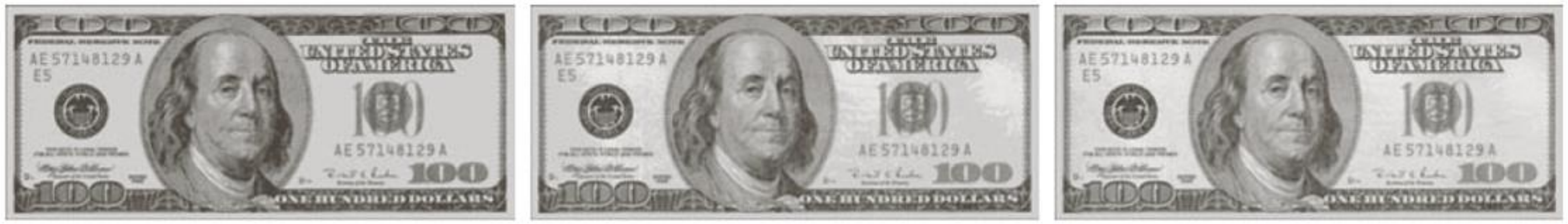
a b c
d e f
g h i

(a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

Course 4

The binary image for the 8-th bit plane of an 8-bit image can be obtained by processing the input image with a threshold intensity transformation function that maps all the intensities between 0 and 127 to 0 and maps all levels between 128 and 255 to 1.

The bit-slicing technique is useful for analyzing the relative importance of each bit in the image – helps in determining the proper number of bits to use when quantizing the image. The technique is also useful for image compression.



a b c

Images reconstructed using (a) bit planes 8 and 7; (b) bit planes 8, 7, and 6; and (c) bit planes 8, 7, 6, and 5.

Course 4

Histogram processing

The *histogram* of a digital image is with intensity levels in ***[0, L-1]***:

$$h(r_k) = n_k \ , \ k = 0,1,\dots,L-1$$

r_k the k-th intensity level

n_k the number of pixels in the image with intensity r_k

Normalized histogram for an ***M x N*** digital image:

$$p(r_k) = \frac{n_k}{MN} \quad , \ k = 0,1,\dots,L-1$$

$p(r_k)$ = an estimate of the probability of occurrence of intensity level r_k in the image

$$\sum_{k=0}^{L-1} p(r_k) = 1$$

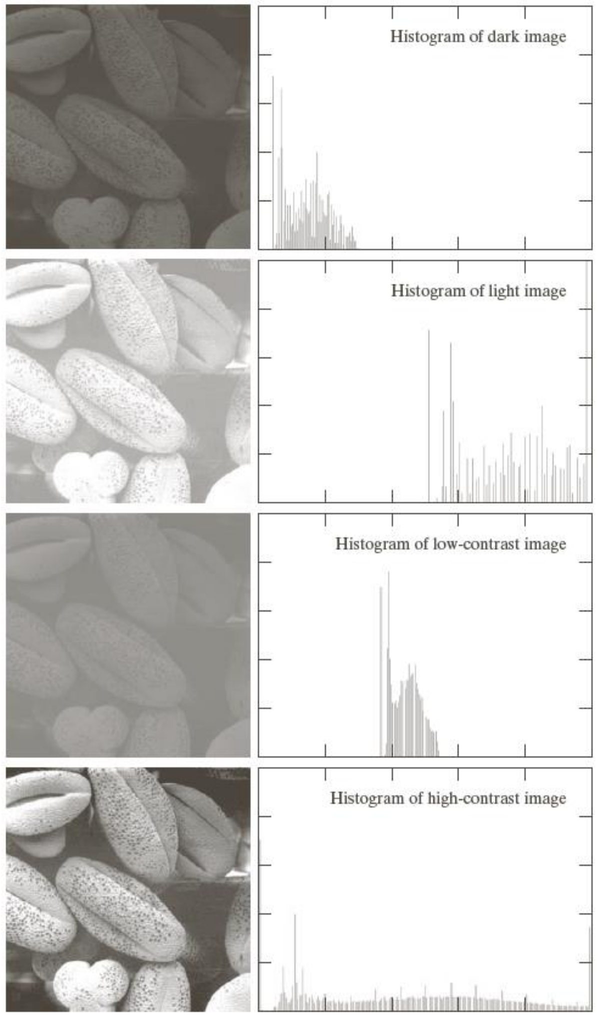


Fig. 8 – dark and light images, low-contrast, and high-contrast images and their histograms

Histogram Equalization (example)

- determine a transformation function that seeks to produce an output image that has a uniform histogram

$$s = T(r) \quad , \quad 0 \le r \le L-1$$

1. ***T(r)*** – monotonically increasing
2. $0 \le T(r) \le L-1$ for $0 \le r \le L-1$

T(r) monotonically increasing – guarantees that intensity values in output image will not be less than the corresponding input values

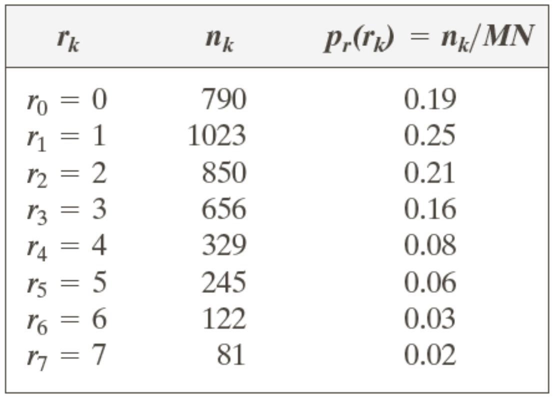
Relation (b) requires that both input and output images have the same range of intensities

Histogram equalization or *histogram linearization* transformation

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

The output image is obtained by mapping each pixel in the input image with intensity r_k into a corresponding pixel with intensity s_k in the output image.

Consider the following example: 3-bit image (***L=8***), ***64x64*** image (***M=N=64, MN=4096***)



r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

Intensity distribution and histogram values for a 3-bit 64×64 digital image

Course 4

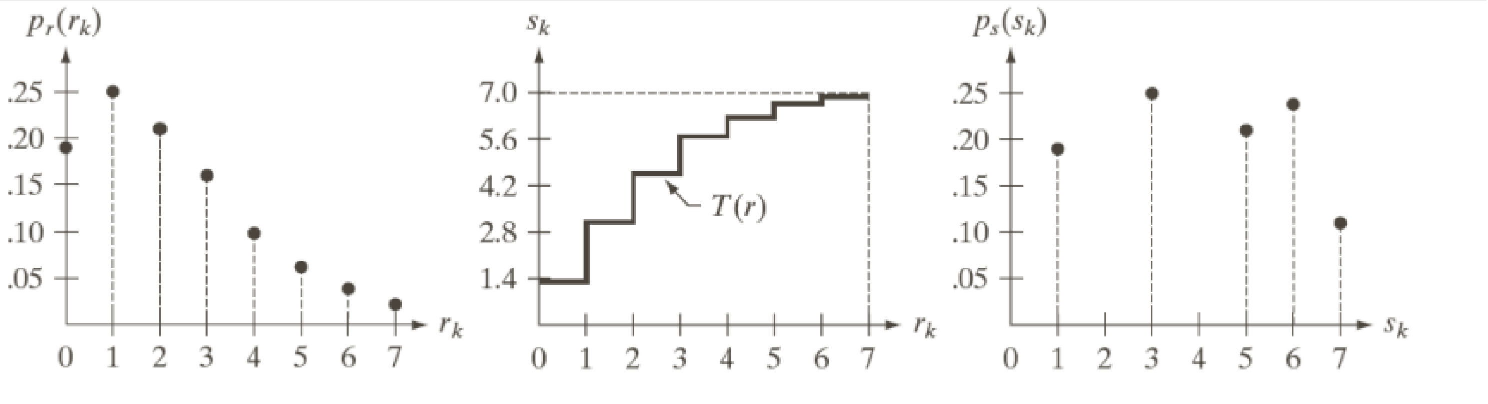
$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

$$s_2 = 4.55 \quad , \quad s_3 = 5.67 \quad , \quad s_4 = 6.23 \quad , \quad s_5 = 6.65 \quad , \quad s_6 = 6.86 \quad , \quad s_7 = 7.00$$

$$\begin{aligned} s_0 &= 1.33 \rightarrow 1 & s_4 &= 6.23 \rightarrow 6 \\ s_1 &= 3.08 \rightarrow 3 & s_5 &= 6.65 \rightarrow 7 \\ s_2 &= 4.55 \rightarrow 5 & s_6 &= 6.86 \rightarrow 7 \\ s_3 &= 5.67 \rightarrow 6 & s_7 &= 7.00 \rightarrow 7 \end{aligned}$$

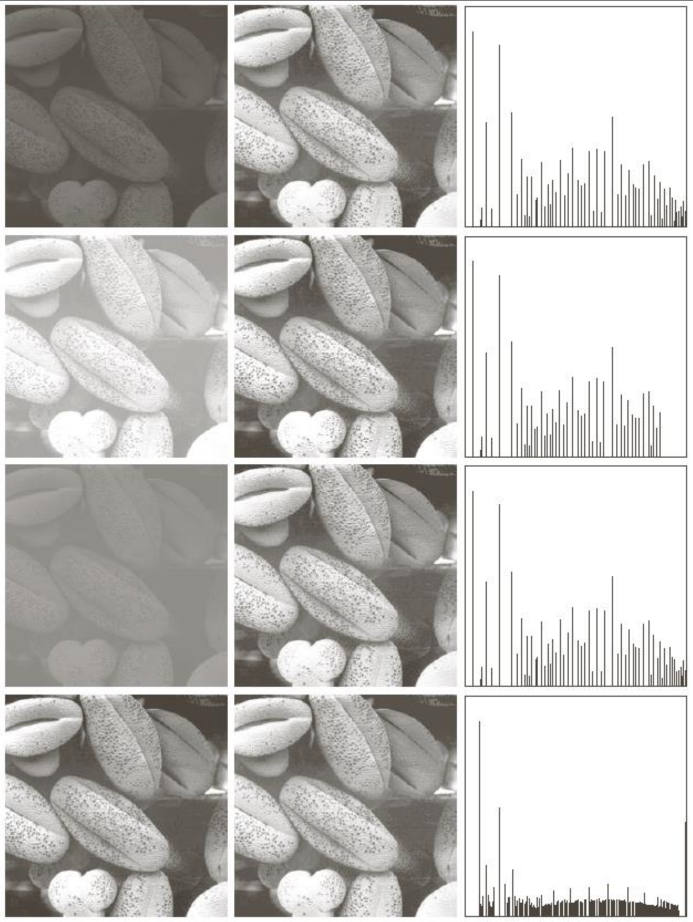
Course 4

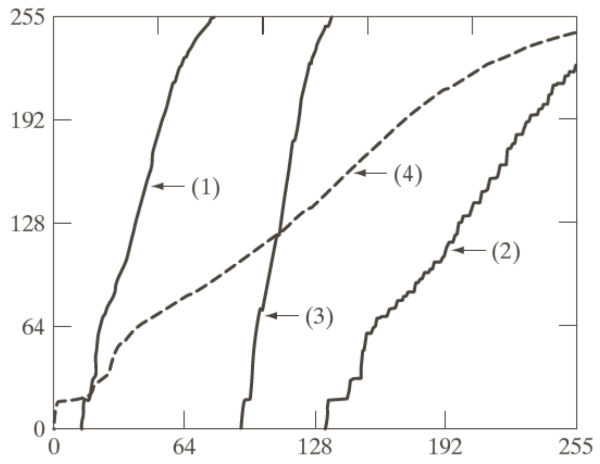


a b c

Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

Course 4





Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images

Course 4

Histogram Matching (Specification) (example)

Sometimes is useful to be able to specify the shape of the histogram that we wish the output image to have. The method used to generate a processed image that has a specified histogram is called *histogram matching* or *histogram specification*.

Suppose ***{z_q; q=0,…,L-1}*** are the new values of histogram we desire to match.

Consider the histogram equalization transformation for the input image:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{M \cdot N}\sum_{j=0}^{k} n_j \ , \quad k = 0,1,\dots,L-1 \qquad (1)$$

Consider the histogram equalization transformation for the new histogram:

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) \ , \quad q = 0,1,\dots,L-1 \qquad (2)$$

$$T(r_k) = s_k = G(z_q) \quad \text{for some value of } q$$

$$z_q = G^{-1}(s_k)$$

Course 4

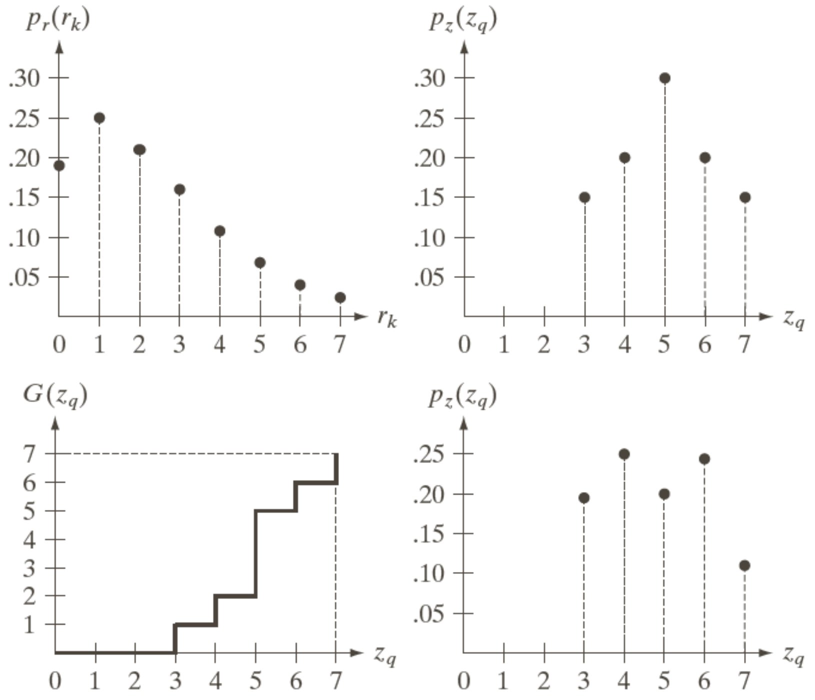
Histogram-specification procedure:

1) Compute the histogram $p_r(r)$ of the input image, and compute the histogram equalization transformation (1). Round the resulting values s_k to integers in $[0, L-1]$
2) Compute all values of the transformation function G using relation (2), where $p_z(z_i)$ are the values of the specified histogram. Round the values $G(z_q)$ to integers in the range $[0, L-1]$ and store these values in a table
3) For every value of s_k ,$k=0,1,\dots,L-1$ use the table for the values of G to find the corresponding value of z_q so that $G(z_q)$ is closest to s_k and store these mappings from s to z. When more than one value of z_q satisfies the property (i.e., the mapping is not unique), choose the smallest value by convention.
4) Form the histogram-specified image by first histogram-equalizing the input image and then mapping every equalized pixel value, s_k , of this image to the corresponding value z_q in the histogram-specified image using the mappings found at step 3).

Course 4

The intermediate step of equalizing the input image can be skipped by combining the two transformation functions ***T*** and $\boldsymbol{G}^{-1}$.

Reconsider the above example:



a b
c d

(a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

Fig. 9

Course 4

Figure 9(a) shows the histogram of the original image. Figure 9 (b) is the new histogram to achieve.

The first step is to obtain the scaled histogram-equalized values:

$$\begin{array}{ll} s_0 = 1 & s_4 = 6 \\ s_1 = 3 & s_5 = 7 \\ s_2 = 5 & s_6 = 7 \\ s_3 = 6 & s_7 = 7 \end{array}$$

Then we compute the values of ***G***:

$$G(z_0) = 7\sum_{i=0}^{0} p_z(z_i) = 0.00 \ \ , \ \ G(z_1) = G(z_2) = 0.00 \ \ , \ \ G(z_3) = 1.05 \rightarrow 1$$

$$G(z_4) = 2.45 \rightarrow 2 \ \ , \ \ G(z_5) = 4.55 \rightarrow 5, \ \ G(z_6) = 5.95 \rightarrow 6 \ \ , \ \ G(z_7) = 7.00 \rightarrow 7$$

z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

The results of performing step 3) of the procedure are summarized in the next table:

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

In the last step of the algorithm, we use the mappings in the above table to map every pixel in the histogram equalized image into a corresponding pixel in the newly-created histogram-specified image. The values of the resulting histogram are listed in the third column of Table 3.2, and the histogram is sketched in Figure 9(d).

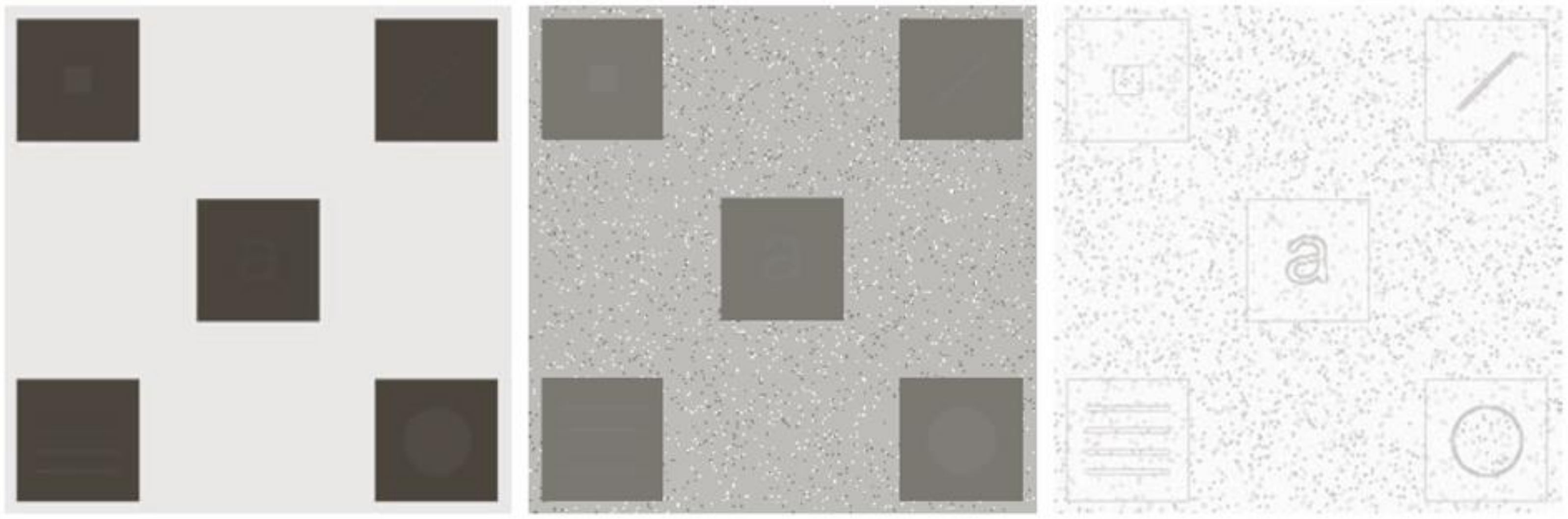
$r_0 = 0$	790
$r_1 = 1$	1023
$r_2 = 2$	850
$r_3 = 3$	656
$r_4 = 4$	329
$r_5 = 5$	245
$r_6 = 6$	122
$r_7 = 7$	81

$s_0 = 1$	790	$\rightarrow z_q = 3$
$s_1 = 3$	1023	$\rightarrow z_q = 4$
$s_2 = 5$	850	$\rightarrow z_q = 5$
$s_3 = s_4 = 6$	$656 + 329$	$\rightarrow z_q = 6$
$s_5 = s_6 = s_7 = 7$	$245 + 122 + 81$	$\rightarrow z_q = 7$

Local Histogram Processing

The histogram processing techniques previously described are easily adaptable to local enhancement. The procedure is to 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 a histogram equalization or histogram specification transformation function is obtained. This function is finally used to map the gray level of the pixel centered in the neighborhood. The center of the neighborhood region is then moved to an adjacent pixel location and the procedure is repeated. Updating the histogram obtained in the previous location with the new data introduced at each motion step is possible.

Course 4



a b c

(a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3 × 3.

Using Histogram Statistics for Image Enhancement

Let ***r*** denote a discrete random variable representing discrete gray-levels in ***[0, L-1]***, and let ***p(r*** $p(r_i)$ denote the normalized histogram component corresponding to the ***i***-th value of ***r***. The ***n***-th moment of ***r*** about its mean is defined as:

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n \, p(r_i)$$

m is the mean (average intensity) value of ***r***:

$$m = \sum_{i=0}^{L-1} r_i \, p(r_i) \quad \text{- measure of average intensity}$$

$$\mu_2(r) = \sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2 \, p(r_i) \ , \ \ \sigma - \ \text{measure of contrast}$$

Sample mean and *sample variance*:

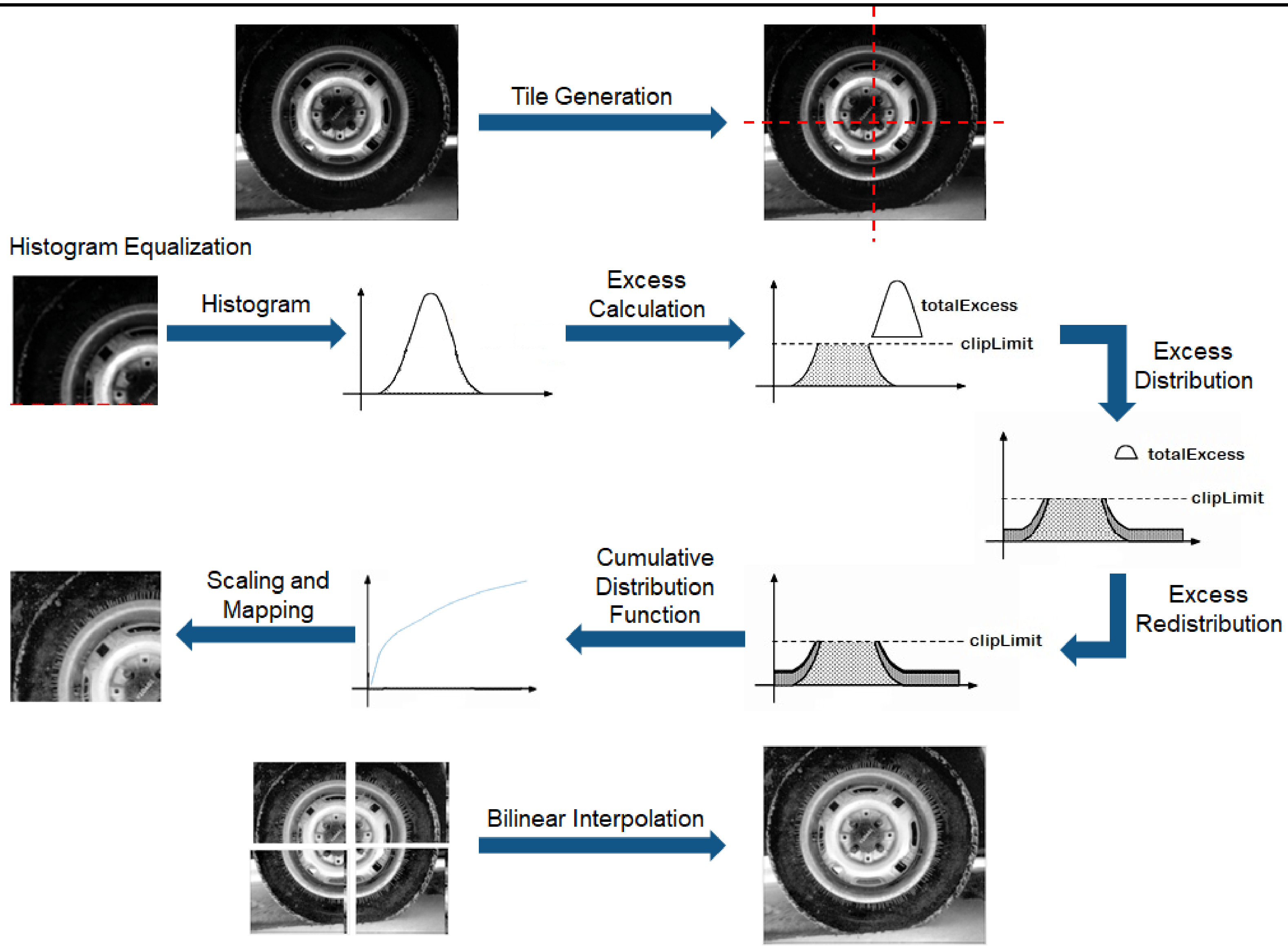
$$m = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \quad , \quad \sigma^2 = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[f(x,y) - m \right]^2$$

CLAHE – Contrast Limited Adaptive Histogram Equalization

https://ww2.mathworks.cn/help/visionhdl/ug/contrast-adaptive-histogram-equalization.html

1. tile generation – division of image in non-overlapping $m \times n$ regions
2. histogram equalization
   - i) histogram computation,
   - ii) excess calculation,
   - iii) excess distribution,
   - iv) excess redistribution,
   - v) scaling and mapping using a cumulative distribution function (CDF)
3. bilinear interpolation – stitching the new tiles

Course 4


Tile Generation
Histogram Equalization
Histogram
Excess Calculation
totalExcess
clipLimit
Excess Distribution
totalExcess
clipLimit
Excess Redistribution
clipLimit
Cumulative Distribution Function
Scaling and Mapping
Bilinear Interpolation

Spatial Filtering

The name *filter* is borrowed from frequency domain processing, where '*filtering*' means accepting (passing) or rejecting certain frequency components. Filters that pass low frequency are called *lowpass* filters. A lowpass filter has the effect of blurring (smoothing) an image. The *filter*s are also called *masks*, *kernels*, *templates* or *windows*.

The Mechanics of Spatial Filtering

A spatial filter consists of:

1) a *neighborhood* (usually a small rectangle)
2) a *predefined operation* performed on the pixels in the neighborhood

Filtering creates a new pixel with the same coordinates as the pixel in the center of the neighborhood, and whose intensity value is modified by the filtering operation.

If the operation performed on the image pixels is linear, the filter is called *linear spatial filter*, otherwise the filter is *nonlinear*.

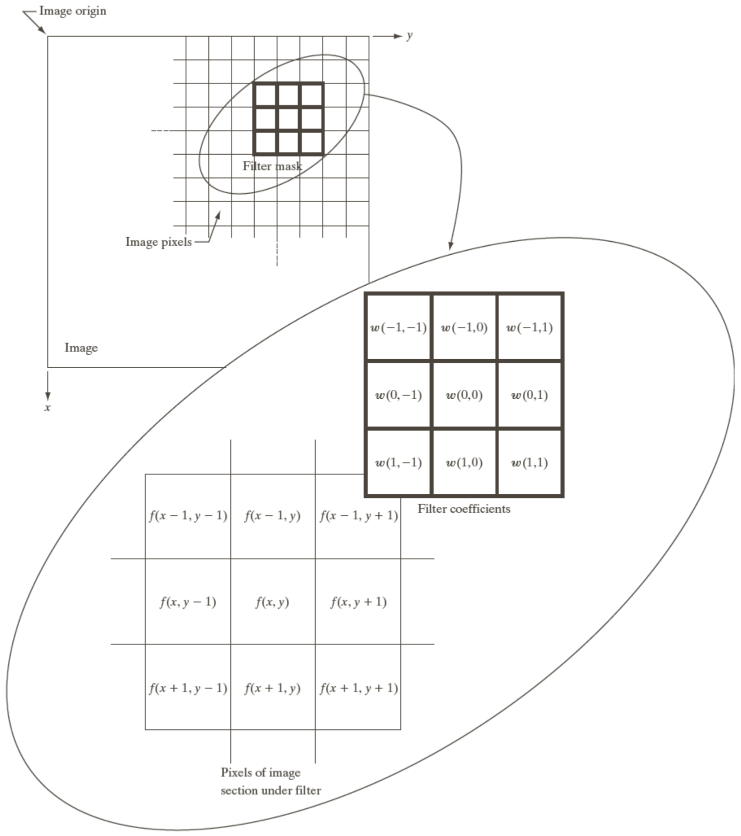


Fig. 10 – Linear spatial filtering with a $\boldsymbol{3 \times 3}$ filter mask

Course 4

In Figure 10 is pictured a ***3* × *3*** linear filter:

$$g(x,y) = w(-1,-1)\,f(x-1,y-1) + w(-1,0)\,f(x-1,y) + \cdots + \\ + w(0,0)\,f(x,y) + \cdots + w(1,1)\,f(x+1,y+1)$$

For a mask of size ***m*× *n***, we assume ***m=2a+1*** and ***n=2b+1***, where ***a*** and ***b*** are positive integers. The general expression of a linear spatial filter of an image of size ***M* × *N*** with a filter of size ***m*× *n*** is:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\,f(x+s,y+t)$$

Smoothing Linear Filters

A smoothing linear filter computes the average of the pixels contained in the neighborhood of the filter mask. These filters are called sometimes *averaging filters* or *lowpass filters*.

The process of replacing the value of every pixel in an image by the average of the intensity levels in the neighborhood defined by the filter mask produces an image with reduced “sharp” transitions in intensities. Usually random noise is characterized by such sharp transitions in intensity levels $\rightarrow$ smoothing linear filters are applied for noise reduction. The problem is that edges are also characterized by sharp intensity transitions, so averaging filters have the undesirable effect that they blur edges.

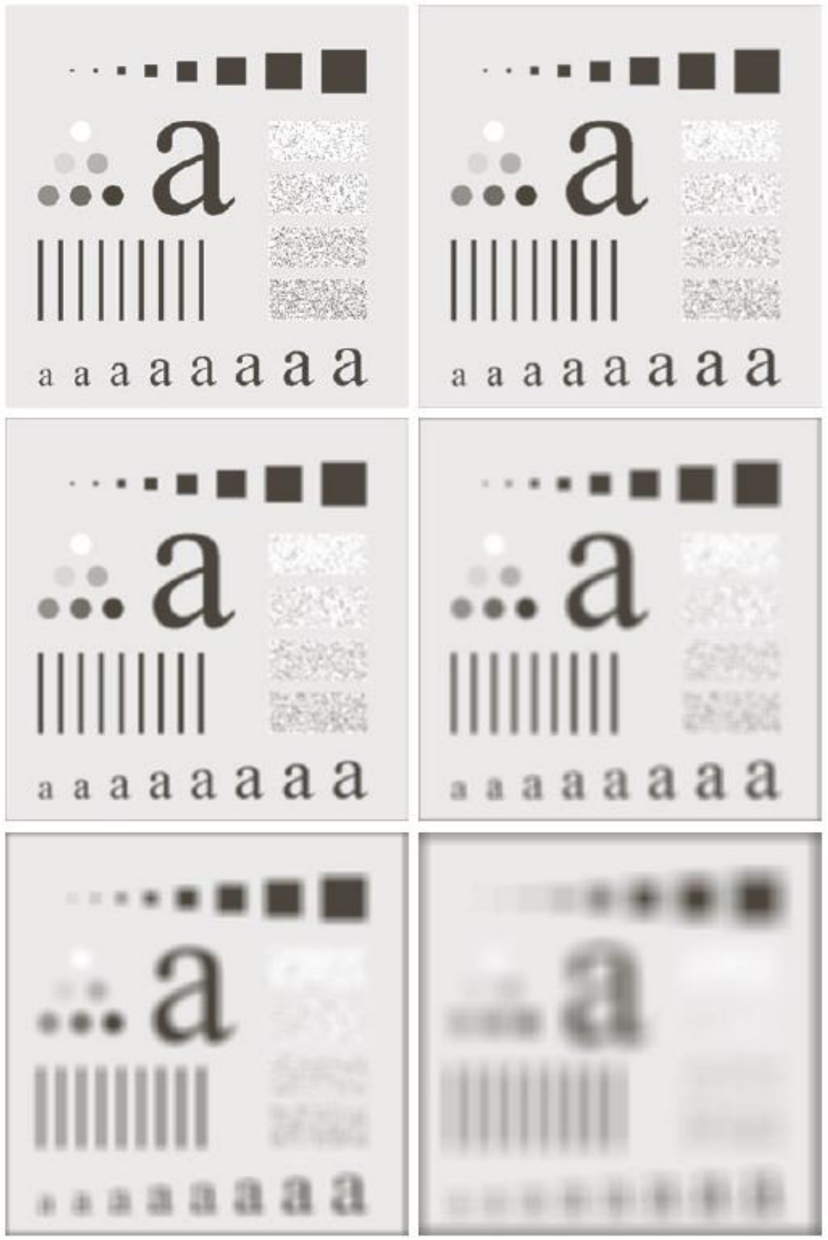
A major use of averaging filters is the reduction of “irrelevant” detail in an image (pixel regions that are small with respect to the size of the filter mask).

There is the possibility of using *weighted average*: the pixels are multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of other.

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array} \qquad \frac{1}{16}\times\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

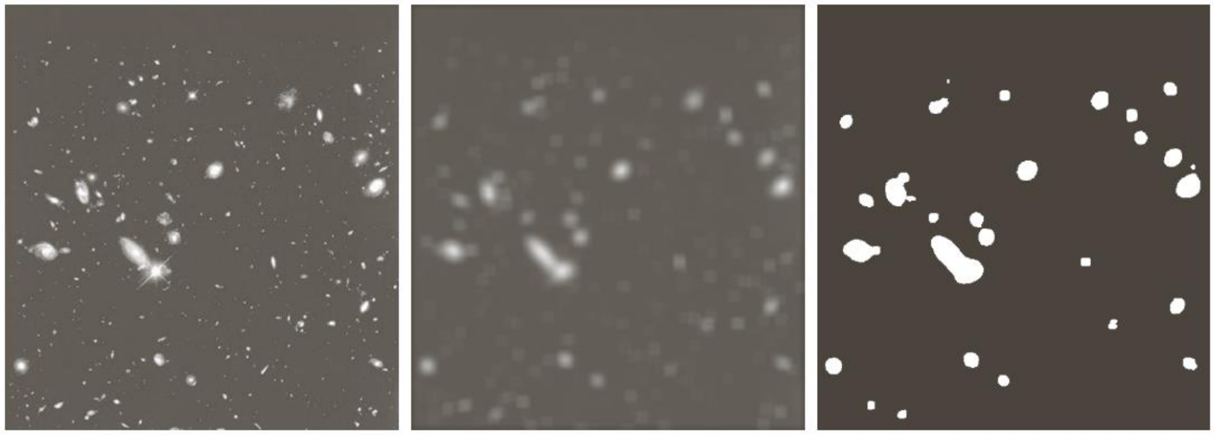
A general weighted averaging filter of size $m\times n$ (m and n are odd) for an $M\times N$ image is given by the expression:

$$g(x,y)=\frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\,f(x+s,y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)} \qquad x=0,1,\dots,M-1\ ,\ y=0,1,\dots,N-1$$



a b
c d
e f
(a) – original image 500×500
(b) – (f) – results of smoothing with square averaging filters of size *m*=3,5,9,15, and 35, respectively
The black squares at the top are of size 3, 5, 9, 15, 25, 35, 45, 55. The letters at the bottom range in size from 10 cu 24 points. The vertical bars are 5 pixels wide and 100 pixels high, separated bu 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart. The noisy rectangles are 50×120 pixels.

An important application of spatial averaging is to blur an image for the purpose of getting a gross representation of objects of interest, such that the intensity of smaller objects blends with the background and larger object become “blob like” and easy to detect. The size of the mask establishes the relative size of the objects that will “disappear” in the background.



Left – image from the Hubble Space Telescope, 528×485; Middle – Image filtered with a 15×15 averaging mask; Right – result of thresholding the middle image

Order-Statistic (Nonlinear) Filters

Order-statistic filters are nonlinear spatial filters based on ordering (ranking) the pixels contained in the image area defined by the selected neighborhood and replacing the value of the center pixel with the value determined by the ranking result. The best known filter in this class is the *median filter*, which replaces the value of a pixel by the median of the intensity values in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median). Median filters provide excellent noise-reduction capabilities, and are less blurring than linear smoothing filters of similar size. Median filters are particularly effective against *impulse noise* (also called *salt-and-pepper noise*).

The *median* ξ, of a set of values is such that half the values in the set are less than or equal to ξ, and half are greater than or equal to ξ.

Course 4

For a $\boldsymbol{3\times 3}$ neighborhood with intensity values ***(10, 15, 20, 20, 30, 20, 20, 25, 100)*** the median is $\boldsymbol{\xi=20}$.

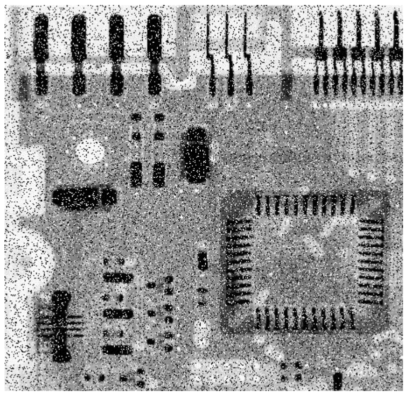
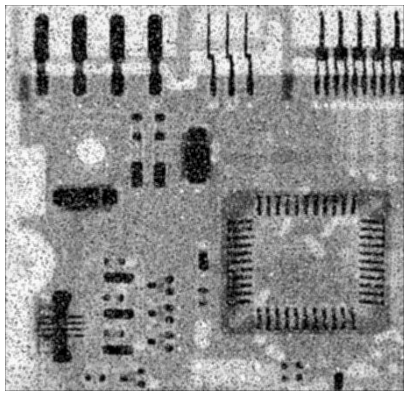
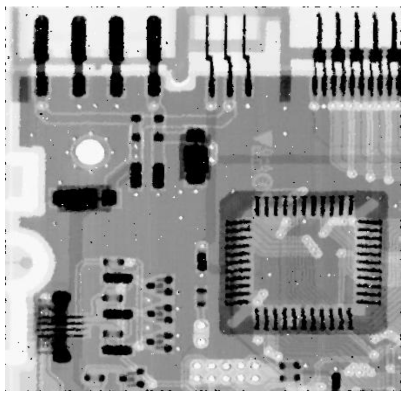
The effect of median filter is to force points with distinct intensity levels to be more like their neighbors. Isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $\boldsymbol{\frac{m^2}{2}}$ are eliminated by an $\boldsymbol{m\times m}$ median filter (eliminated means forced to the median intensity of the neighbors).

Max/min filter is the filter which replaces the intensity value of the pixel with the max/min value of the pixels in the neighborhood. The max/min filter is useful for finding the brightest/darkest points in an image.

Min filter – 0% filter

Median filter – 50% filter

Max filter – 100% filter

(a) (b) (c)

(a) –X-ray image of circuit board corrupted by salt&pepper noise

(b) – noise reduction with a 3×3 averaging filter

(c) – noise reduction with a 3×3 median filter