Computer Vision

Course 4

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:

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

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

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

Course 4

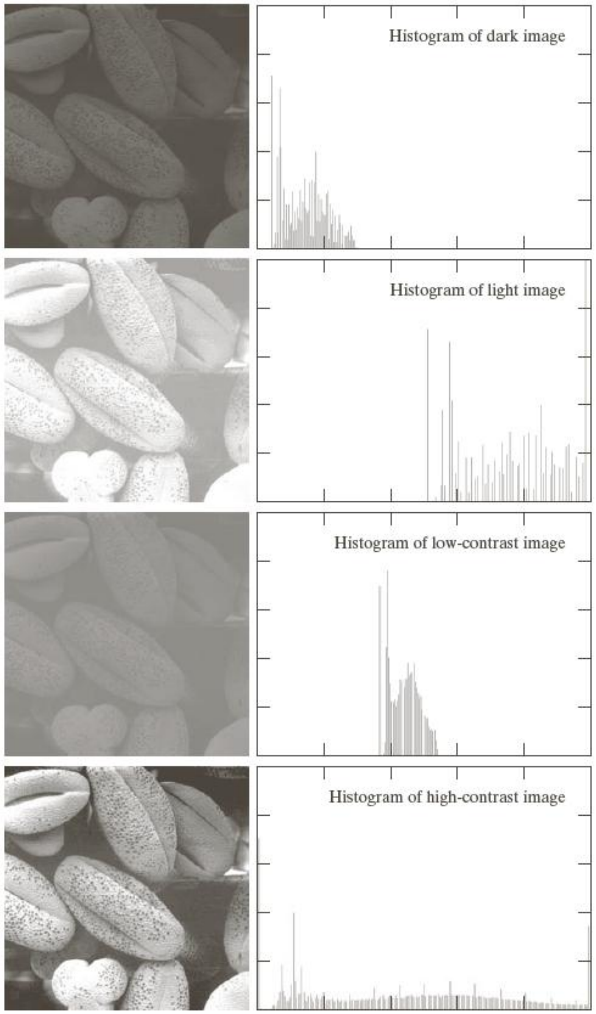


Fig. 8 – dark and light images, low-contrast, and high-contrast images and their histograms

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

Course 4

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

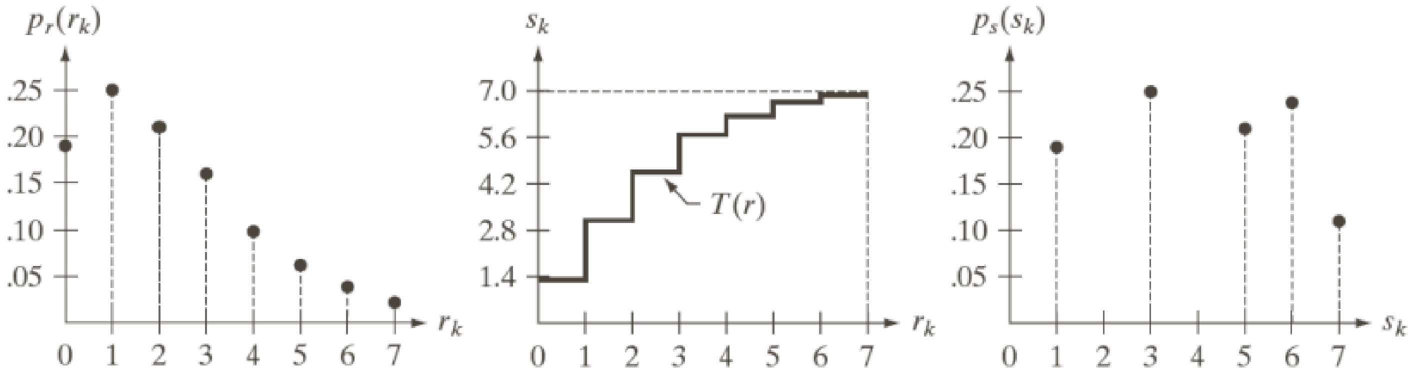
$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

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

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

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

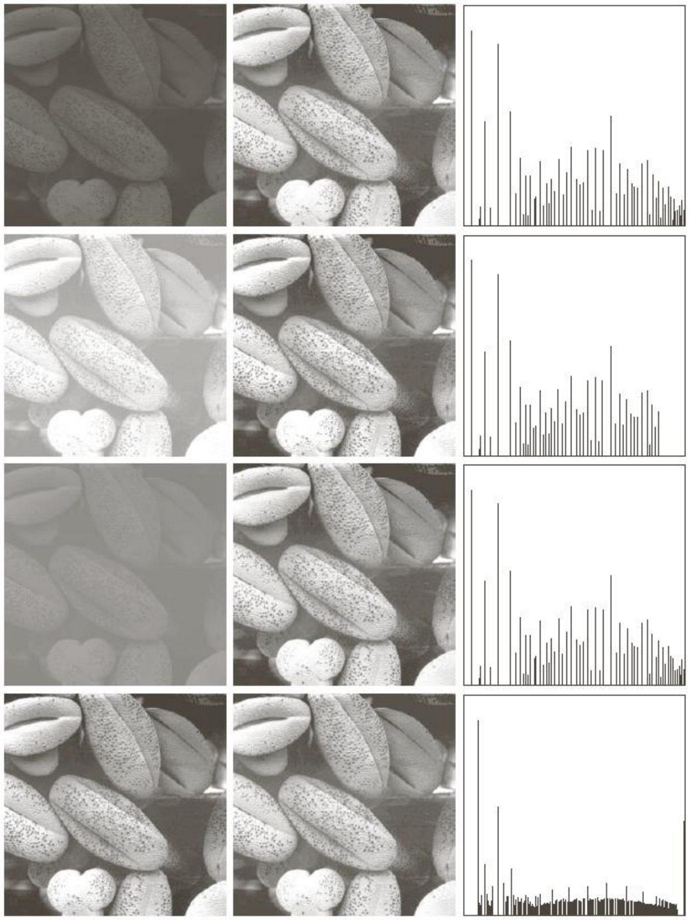
Course 4

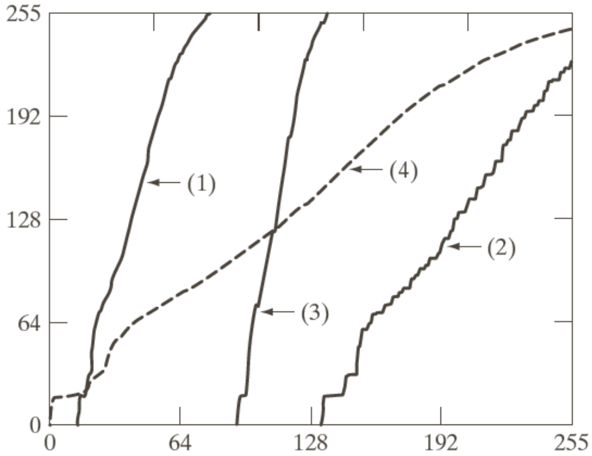


a b c

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

Course 4

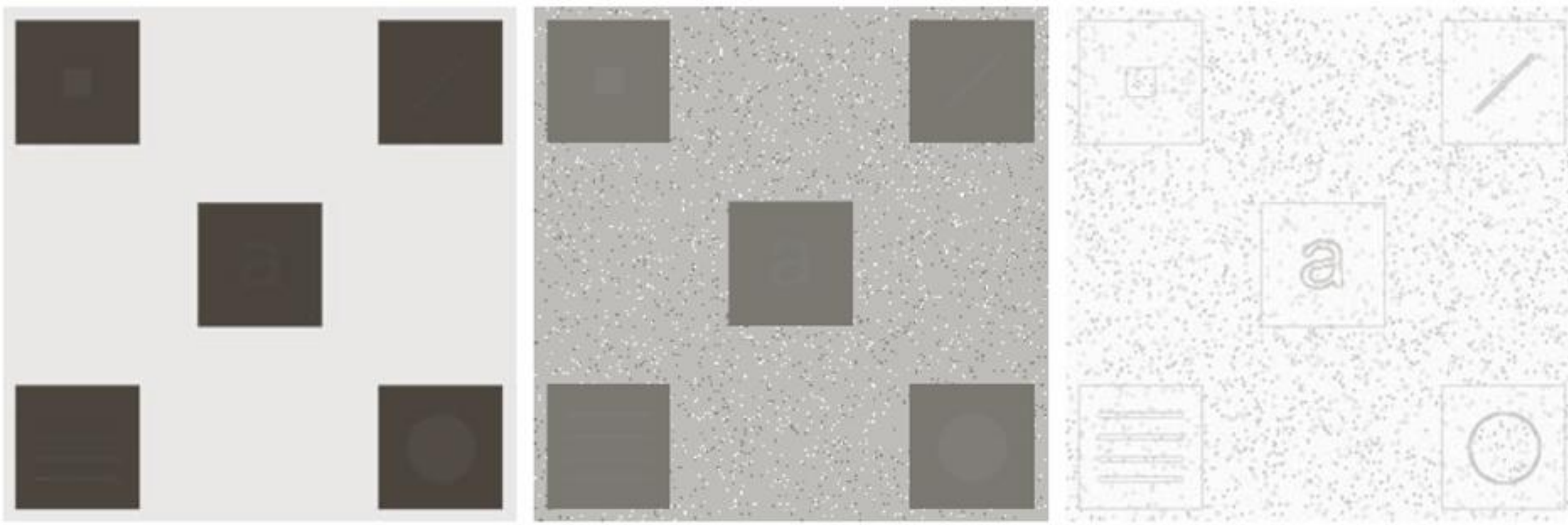




Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images

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.



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.

Course 4

Using Histogram Statistics for Image Enhancement

Let ***r*** denote a discrete random variable representing discrete gray-levels in ***[0, L-1]***, and let $p(r_i)$ 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*:

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

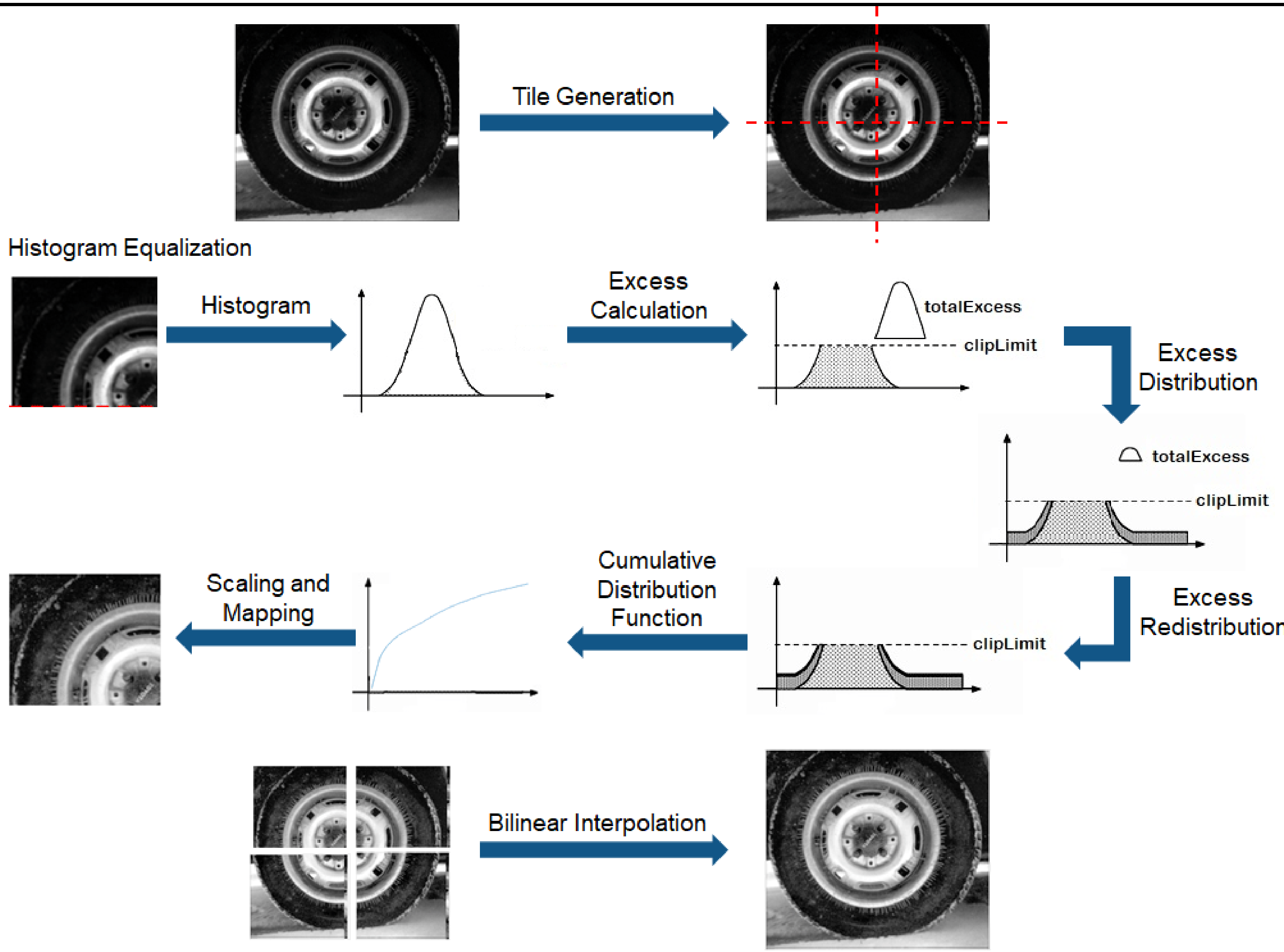
CLAHE – Contrast Limited Adaptive Histogram Equalization

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

1. **Tiles/blocks generation** – division of image in non-overlapping $m \times n$ regions
2. **Computing the histogram for each block**: A pixel intensity histogram is calculated for each block, representing the distribution of brightness levels within that specific region.
3. **Histogram clippin**g: If any histogram bin exceeds a predefined maximum threshold, it is clipped to prevent excessive contrast amplification. The **clipped excess** is then **redistributed** evenly across all histogram bins, reducing noise in uniform areas.

4. **Local histogram equalization**: After adjusting the histogram, histogram equalization is applied to each block. This is done using the cumulative distribution function (CDF), which redistributes pixel intensity levels more uniformly.
5. **Bilinear interpolation between blocks**: To prevent abrupt transitions between blocks when reconstructing the image, bilinear interpolation is applied. This smooths out contrast differences between regions, resulting in a more natural and continuous output.
6. **Merging the processed tiles/blocks**: Finally, after equalization and interpolation, the processed blocks are merged to form the final image with enhanced contrast.

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*.

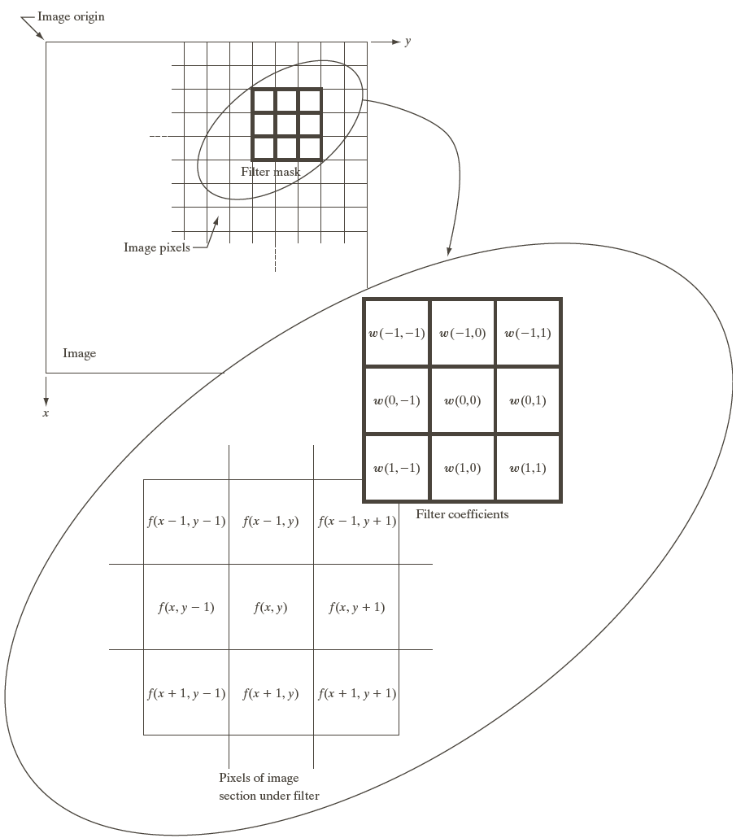


Fig. 10 – Linear spatial filtering with a ***3*** × ***3*** filter mask

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 → 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).

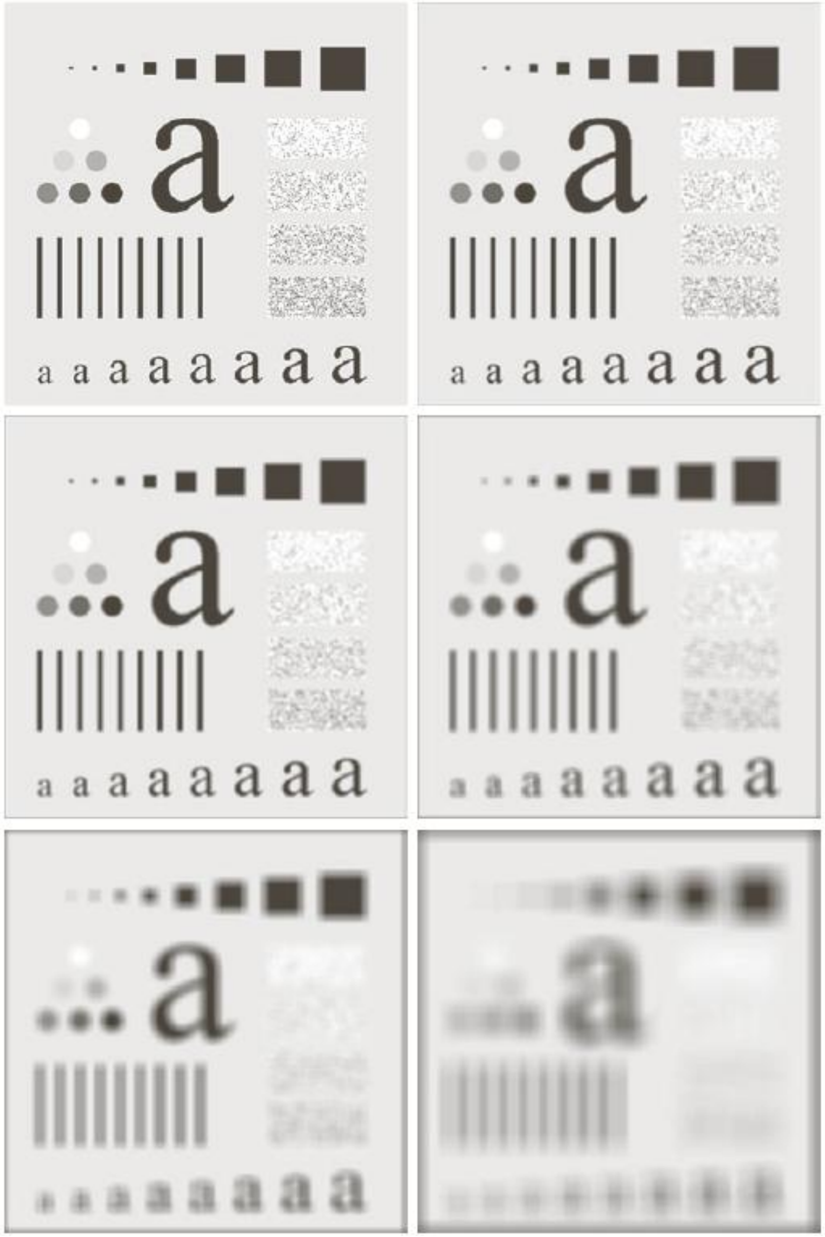
There is the possibility of using *weighted average*: the pixels are multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of other.

$$\frac{1}{9}\times\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \frac{1}{16}\times\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

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:

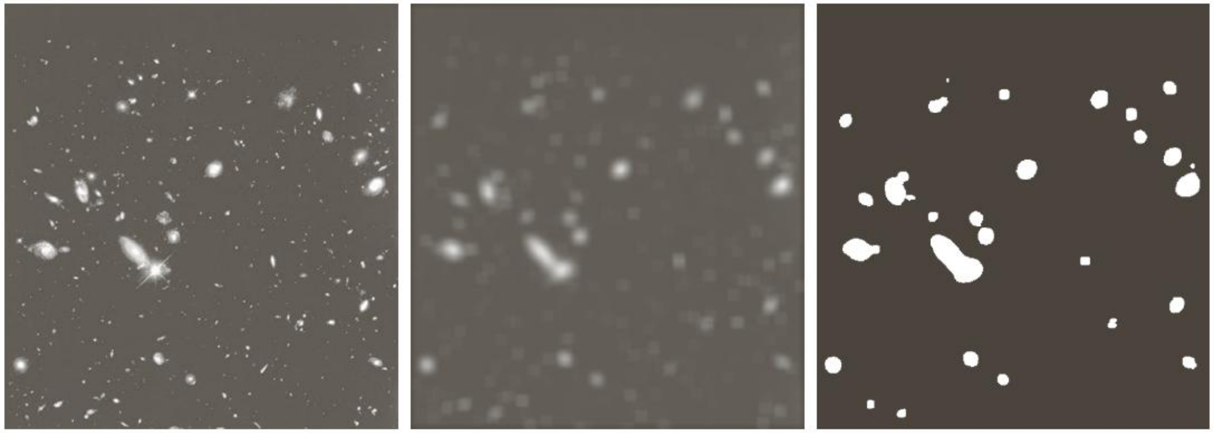
$$g(x,y)=\frac{\displaystyle\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\, f(x+s,y+t)}{\displaystyle\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)} \qquad x=0,1,\dots,M-1\ ,\ y=0,1,\dots,N-1$$

Course 4



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 about 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 3×3 neighborhood with intensity values ***(10, 15, 20, 20, 30, 20, 20, 25, 100)*** the median is $\xi = 20$.

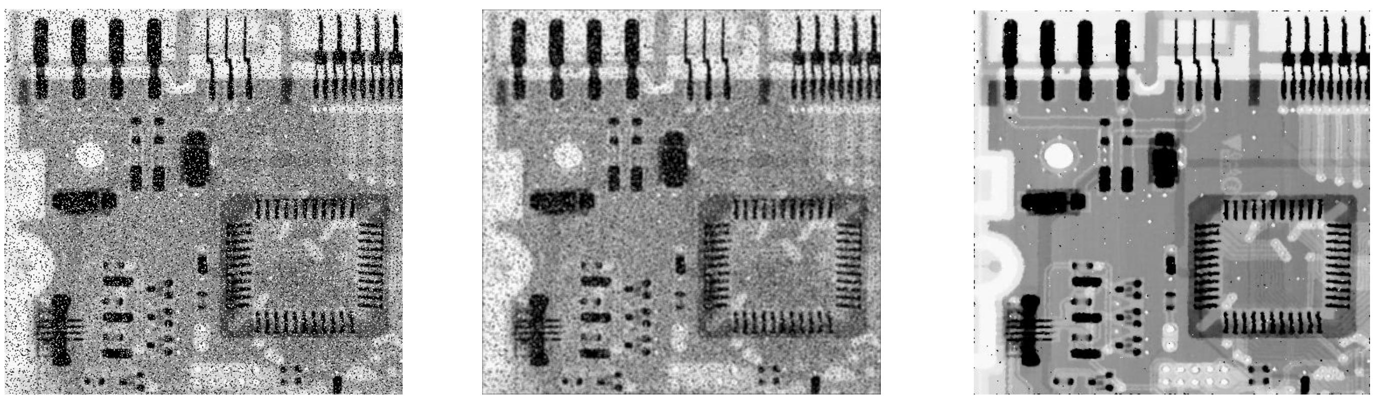
The effect of median filter is to force points with distinct intensity levels to be more like their neighbors. Isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $\frac{m^2}{2}$ are eliminated by an $m\times m$ median filter (eliminated means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

Sharpening Spatial Filters

The principal objective of sharpening is to highlight transitions in intensity. These filters are applied in electronic printing, medical imaging, industrial inspection, autonomous guidance in military systems.

Averaging – analogous to integration

Sharpening – spatial differentiation

Image differentiation enhances edges and other discontinuities (noise, for example) and deemphasizes areas with slowly varying intensities.

For digital images, discrete approximation of derivatives are used

$$\frac{\partial f}{\partial x} \approx f(x+1, y) - f(x, y)$$

$$\frac{\partial^2 f}{\partial x^2} \approx f(x+1, y) - 2f(x, y) + f(x-1, y)$$

Course 4

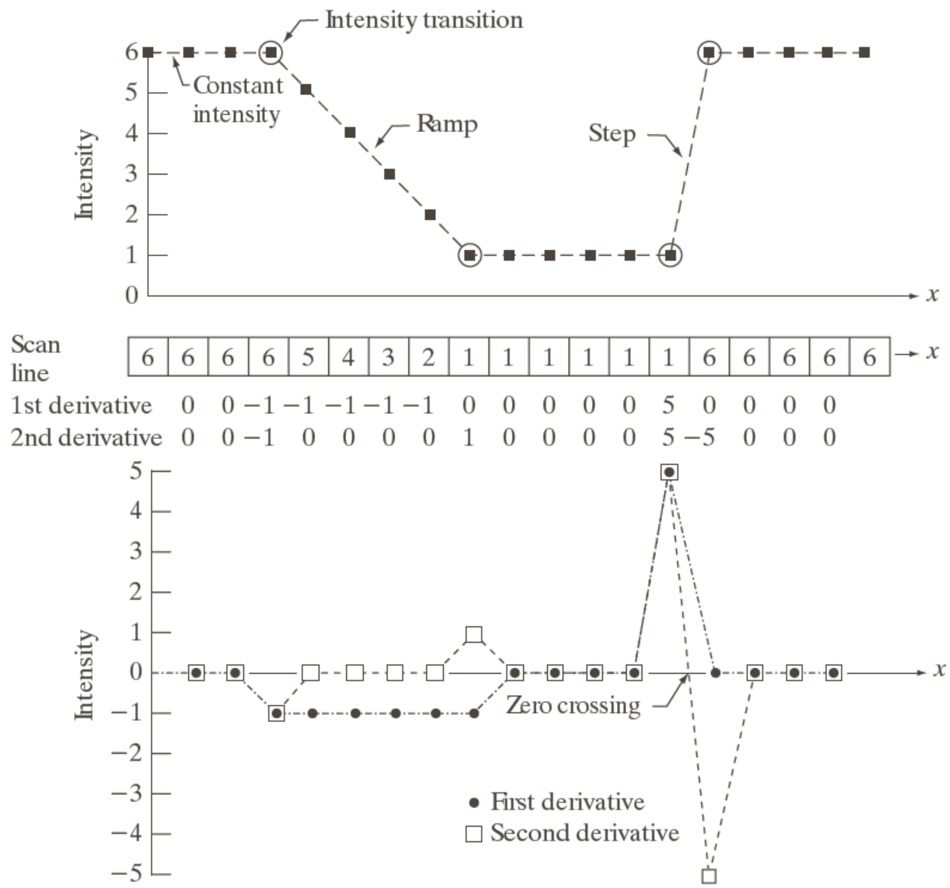


Illustration of the first and second derivatives of a 1-D digital function

Course 4

Using the Second Derivative for Image Sharpening – Laplacian operator

Isotropic filters – the response of this filter is independent of the direction of the discontinuities in the image. Isotropic filters are *rotation invariant*, in the sense that rotating the image and then applying the filter gives the same result as applying the filter to the image and then rotating the result.

The simplest isotropic derivative operator is the Laplacian:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

This operator is linear.

Course 4

$$\frac{\partial^2 f}{\partial x^2} \approx f(x+1, y) - 2f(x, y) + f(x-1, y)$$

$$\frac{\partial^2 f}{\partial y^2} \approx f(x, y+1) - 2f(x, y) + f(x, y-1)$$

$$\nabla^2 f(x, y) \approx f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Filter mask that approximate the Laplacian

The Laplacian being a derivative operator highlights gray-level discontinuities in an image and deemphasizes regions with slowly varying gray levels. This will tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a dark, featureless background. Background features can be "recovered" while still preserving the sharpening effect of the Laplacian operation simply by adding the original and Laplacian images.

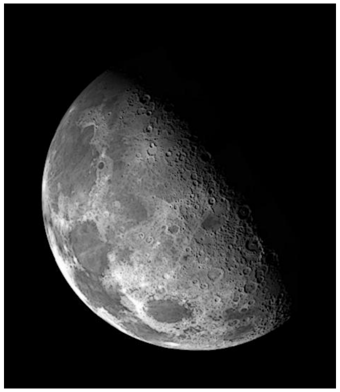
The basic way to use the Laplacian for image sharpening is given by:

$$g(x,y) = f(x,y) + c\nabla^2 f(x,y)$$

The (discrete) Laplacian can contain both negative and positive values – it needs to be scaled.

Course 4

Blurred image of the North Pole of the Moon; Laplace filtered image



Sharpening with *c=1* and *c=2*

Unsharp Masking and Highboost Filtering

- process used in printing and publishing industry to sharpen images
- subtracting an unsharp (smoothed) version of an image from the original image

1. Blur the original image
2. Subtract the blurred image from the original (the resulting difference is called the mask)
3. Add the mask to the original

Let $\bar{f}(x, y)$ be the blurred image. The mask is given by:

$$g_{\text{mask}}(x, y) = f(x, y) - \bar{f}(x, y)$$

$$g(x, y) = f(x, y) + k \cdot g_{\text{mask}}(x, y)$$

k* = *1 – unsharp masking

k > ***1*** – *highboost filtering*

original image

blurred image (Gaussian filter 5×5, σ=3)

mask – difference between the above images

unsharp masking result

highboost filter result ($k=4.5$)

The Gradient for (Nonlinear) Image Sharpening

$$\nabla f \equiv grad(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- the gradient points in the direction of the greatest rate of change of f at location ***(x,y)***.

The *magnitude* (*length*) of the gradient is defined as:

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

M(x,y) is an image of the same size as the original called the *gradient image* (or simply as the *gradient*). ***M(x,y)*** is rotation invariant (isotropic) (the gradient vector ∇f is not isotropic). In some applications, the following formula is used:

$$M(x,y) \approx |g_x| + |g_y| \quad \text{(not isotropic)}$$

Different ways of approximating g_x and g_y produce different filter operators.

Roberts cross-gradient operator (1965)

$$g_x \approx f(x+1, y+1) - f(x,y) = \Delta_1$$

$$g_y \approx f(x, y+1) - f(x+1, y) = \Delta_2$$

$$M(x,y) = \sqrt{\Delta_1^2 + \Delta_2^2}$$

$$M(x,y) \approx |\Delta_1| + |\Delta_2|$$

Sobel operators

$$g_x \approx \big(f(x-1,y+1) + 2f(x,y+1) + f(x+1,y+1) \big) - \big(f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1) \big)$$

$$g_y \approx \big(f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1) \big) - \big(f(x-1,y-1) + 2f(x-1,y) + f(x-1,y+1) \big)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts cross gradient operators

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel operators