



3

Removal of Artifacts

Biomedical images are often affected and corrupted by

various types of noise and artifact.

Any image, pattern, or signal other than that of interest

could be termed as interference, artifact, or simply *noise*.



3.1 Characterization of Artifacts

3.1.1 Random noise

Random noise: interference from a random process such as thermal noise in electronic devices and the counting of photons.

Random process characterized by its PDF: probabilities of occurrence of all possible values of a random variable.



Consider a random process η characterized by the PDF $p_\eta(\eta)$.

The process could be a function of time as $\eta(t)$, or of space in 1D, 2D, or 3D as $\eta(x)$, $\eta(x, y)$, or $\eta(x, y, z)$; it could also be a 4D spatio-temporal function as $\eta(x, y, z, t)$.

Mean μ_η : first-order moment of the PDF.

$$\mu_\eta = E[\eta] = \int_{-\infty}^{\infty} \eta p_\eta(\eta) d\eta, \quad (3.1)$$

where $E[]$ represents the *statistical expectation operator*.

Common to assume mean of a random noise process = zero.



Mean-squared (MS) value: second-order moment.

$$E[\eta^2] = \int_{-\infty}^{\infty} \eta^2 p_{\eta}(\eta) d\eta. \quad (3.2)$$

Variance σ_{η}^2 : second central moment.

$$\sigma_{\eta}^2 = E[(\eta - \mu_{\eta})^2] = \int_{-\infty}^{\infty} (\eta - \mu_{\eta})^2 p_{\eta}(\eta) d\eta. \quad (3.3)$$

Square root of variance = standard deviation (SD) σ_{η} .



Note that $\sigma_\eta^2 = E[\eta^2] - \mu_\eta^2$.

If the mean is zero, it follows that $\sigma_\eta^2 = E[\eta^2]$:

variance = MS value.

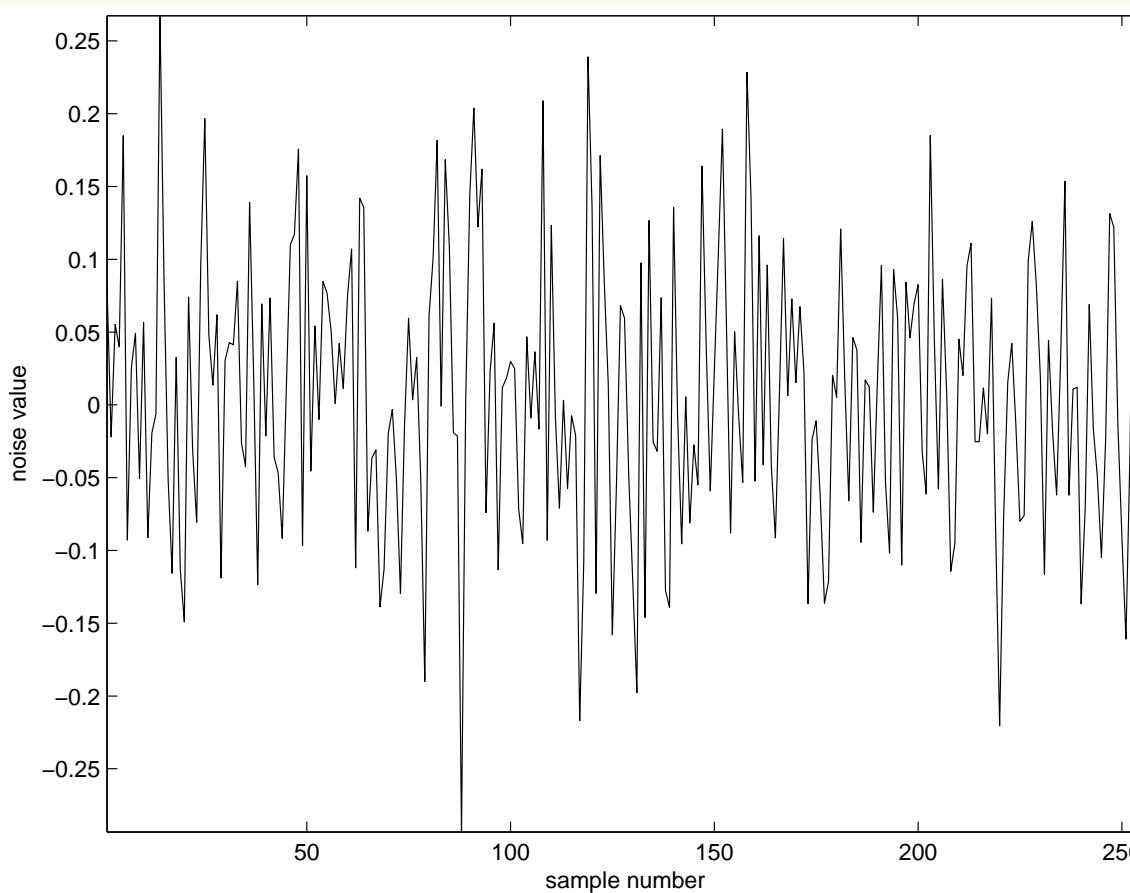


Figure 3.1: A time series composed of random noise samples with a Gaussian PDF having $\mu = 0$ and $\sigma^2 = 0.01$. MS value = 0.01; RMS = 0.1. See also Figures 3.2 and 3.3.

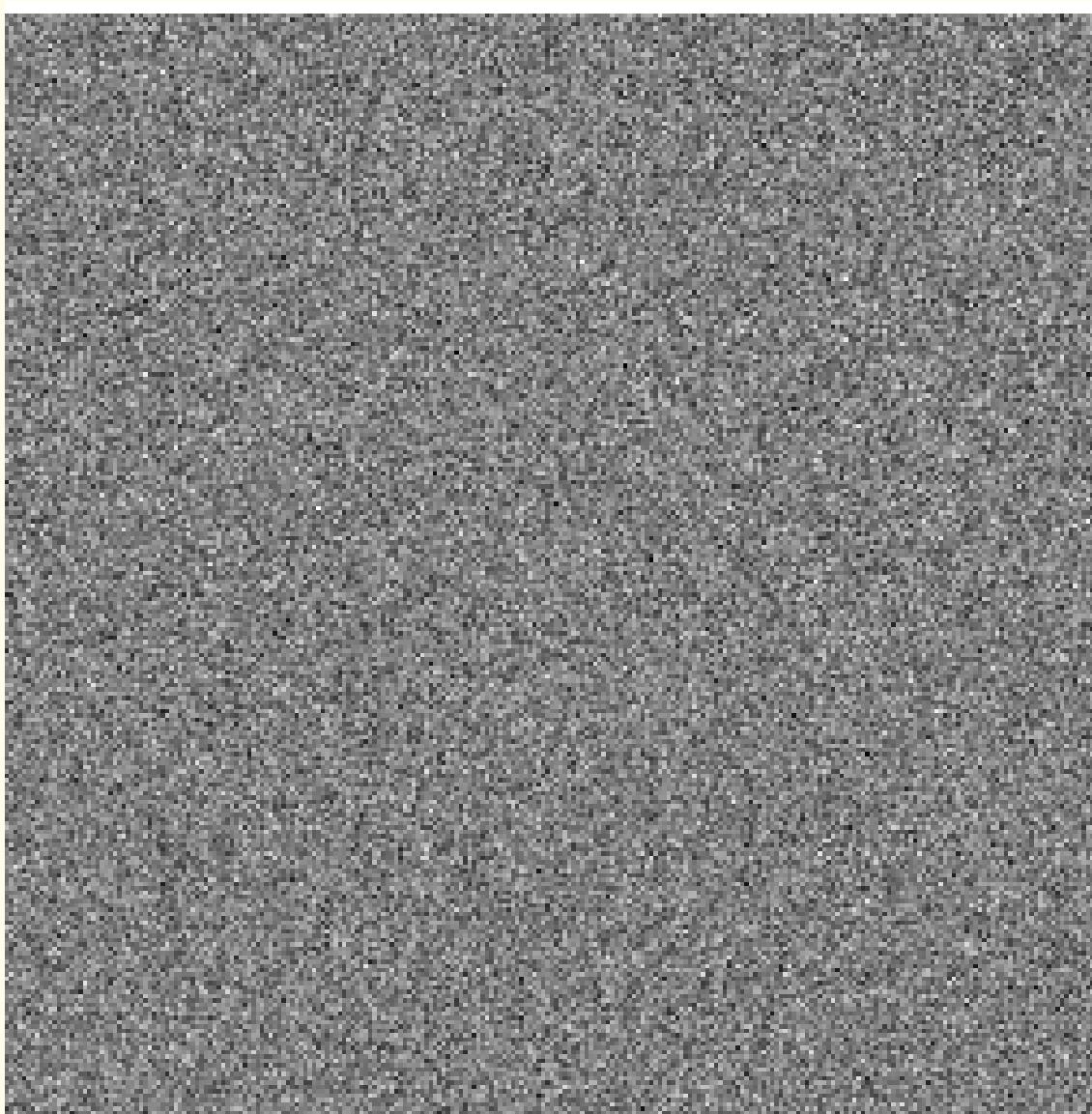


Figure 3.2: An image composed of random noise samples with a Gaussian PDF having $\mu = 0$ and $\sigma^2 = 0.01$. MS value = 0.01; RMS = 0.1. The normalized pixel values in the range $[-0.5, 0.5]$ were linearly mapped to the display range $[0, 255]$. See also Figure 3.3.

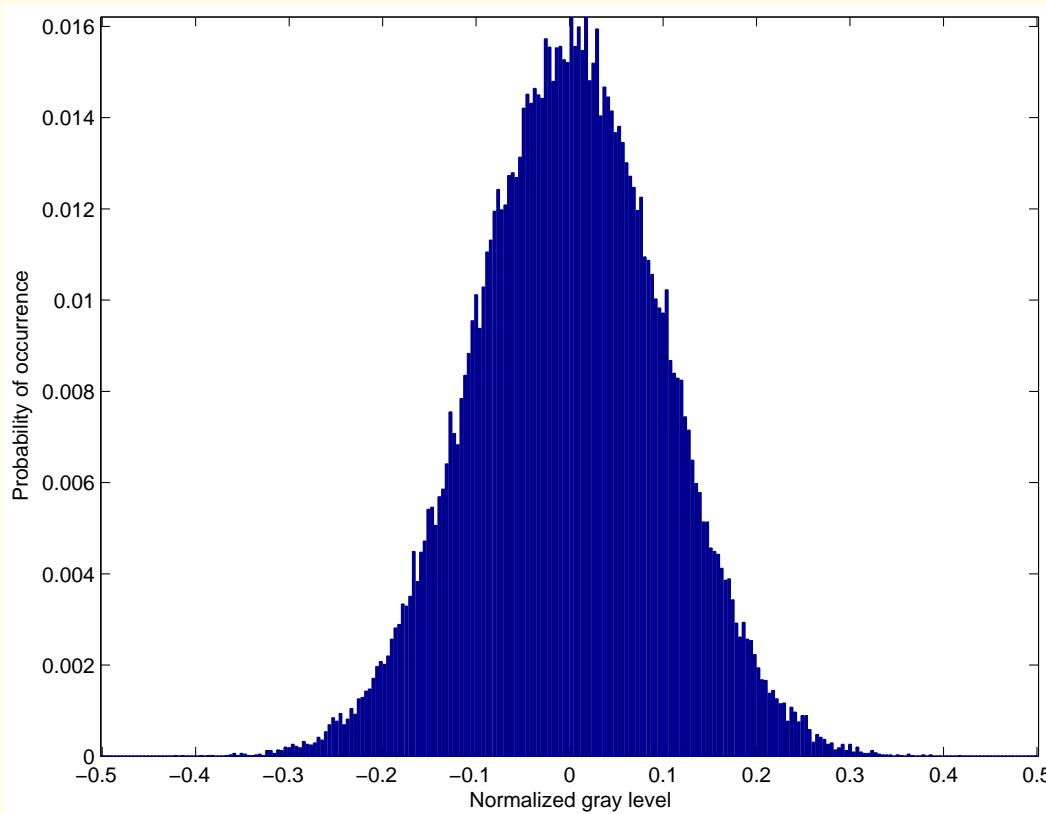


Figure 3.3: Normalized histogram of the image in Figure 3.2. The samples were generated using a Gaussian process with $\mu = 0$ and $\sigma^2 = 0.01$. MS value = 0.01; RMS = 0.1. See also Figures 3.1 and 3.2.



A biomedical image $f(x, y)$ may also, for the sake of generality, be considered to be a *realization of a random process* f .

This allows for the statistical characterization of sample-to-sample or person-to-person variations in

a collection of images of the same organ, system, or type.



Statistical averages representing populations of images of a certain type are useful in

- designing filters,
- data compression techniques, and
- pattern classification procedures

that are optimal for the specific type of images.

However, in diagnostic applications, it is the *deviation from the normal* or the average in the image on hand that is of critical importance.



When an image $f(x, y)$ is observed in the presence of random noise η , the detected image $g(x, y)$ may be treated as a realization of another random process g .

In most cases, the noise is additive:

$$g(x, y) = f(x, y) + \eta(x, y). \quad (3.4)$$

Each of the random processes f , η , and g is characterized by its own PDF $p_f(f)$, $p_\eta(\eta)$, and $p_g(g)$, respectively.



In most practical applications, the random processes representing an image of interest and the noise affecting the image may be assumed to be *statistically independent processes*:

$$p_{f,\eta}(f, \eta) = p_f(f) p_\eta(\eta).$$

$$E[g] = \mu_g = \mu_f + \mu_\eta = \mu_f = E[f], \quad (3.5)$$

$$E[(g - \mu_g)^2] = \sigma_g^2 = \sigma_f^2 + \sigma_\eta^2. \quad (3.6)$$



Ensemble averages:

When the PDFs of the random processes are not known,

approximate the statistical expectation operation by averages computed using a collection or *ensemble* of sample observations.

Suppose we have M observations of the random process f :

$$f_1(x, y), f_2(x, y), \dots, f_M(x, y).$$

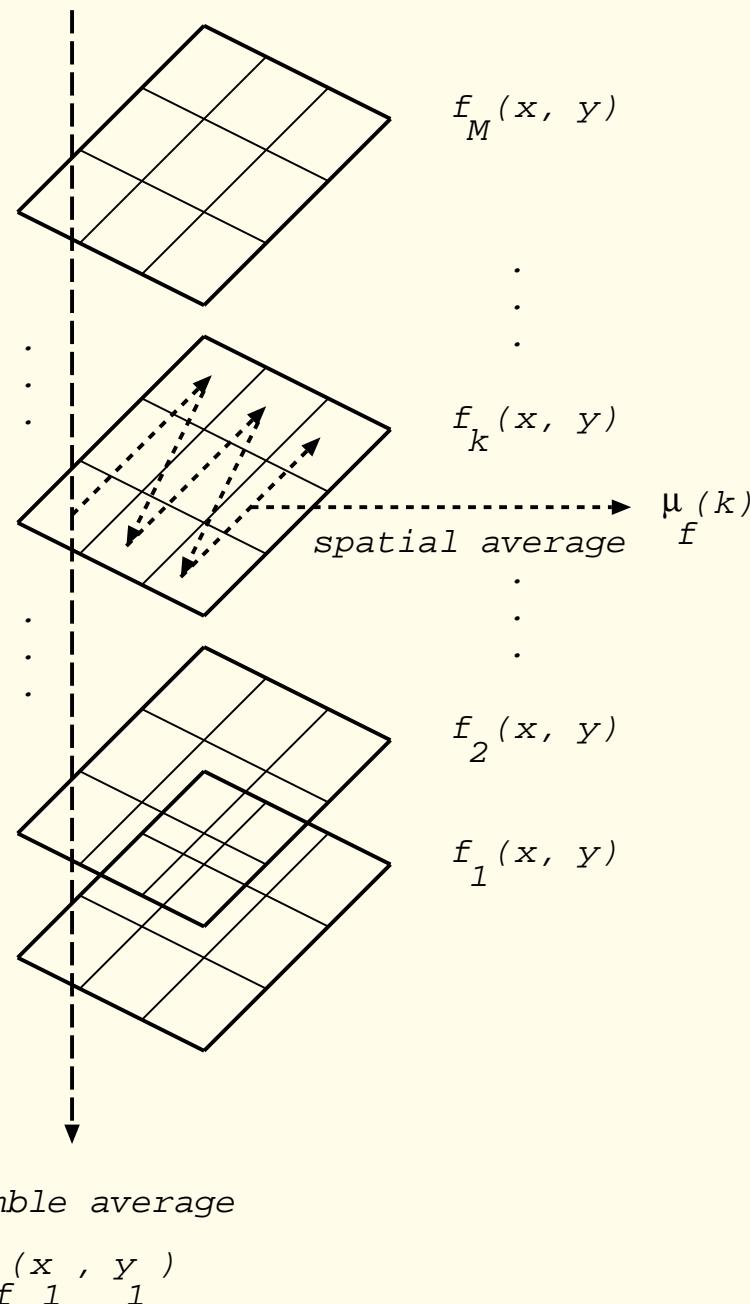


Figure 3.4: Ensemble and spatial averaging of images.



Mean of the process at a particular spatial location (x_1, y_1) :

$$\mu_f(x_1, y_1) = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{k=1}^M f_k(x_1, y_1). \quad (3.7)$$

The autocorrelation function (ACF) defined as

$$\phi_f(x_1, x_1 + \alpha, y_1, y_1 + \beta) = E[f(x_1, y_1), f(x_1 + \alpha, y_1 + \beta)]. \quad (3.8)$$



The ACF may be estimated as

$$\phi_f(x_1, x_1 + \alpha, y_1, y_1 + \beta) = \quad (3.9)$$

$$\lim_{M \rightarrow \infty} \frac{1}{M} \sum_{k=1}^M f_k(x_1, y_1) f_k(x_1 + \alpha, y_1 + \beta).$$

If the process is stationary, the ACF depends only upon the

shift parameters: $\phi_f(\alpha, \beta)$.



When $\mu_f(x_1, y_1)$ is computed for every spatial location or pixel,

we get an average image that could be expressed as $\bar{f}(x, y)$:

may be used to represent the random process f as a prototype.



Temporal and spatial averages:

When we have a sample observation of a random process $f_k(t)$ as a function of time, it is possible to compute *time averages* or *temporal statistics*:

$$\mu_f(k) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} f_k(t) dt. \quad (3.10)$$

Time-averaged ACF:

$$\phi_f(\tau, k) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} f_k(t) f_k(t + \tau) dt. \quad (3.11)$$



Given an observation of a random process as an image $f_k(x, y)$, we may compute *spatial averages* or *spatial statistics*:

Spatial mean of the image $f_k(x, y)$

$$\mu_f(k) = \frac{1}{A} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_k(x, y) dx dy, \quad (3.12)$$

A : normalization factor (e.g., area of the image.)

The spatial mean above is a single-valued entity (a scalar).



For a stationary process, the spatial ACF is given by

$$\phi_f(\alpha, \beta, k) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_k(x, y) f_k(x + \alpha, y + \beta) dx dy. \quad (3.13)$$

Normalization factor: total energy of the image $= \phi_f(0, 0)$.

Sample index k irrelevant if only one observation is available.



When we have a 2D image as a function of time, such as TV, video, fluoroscopy, and cine-angiography signals, we have a spatio-temporal signal that may be expressed as $f(x, y, t)$.

We may then compute statistics over a single frame $f(x, y, t_1)$ at the instant of time t_1 , which are known as *intraframe statistics*.

We could also compute parameters through multiple frames over a certain period of time, which are called *interframe statistics*:

signal over a period of time treated as a 3D dataset.

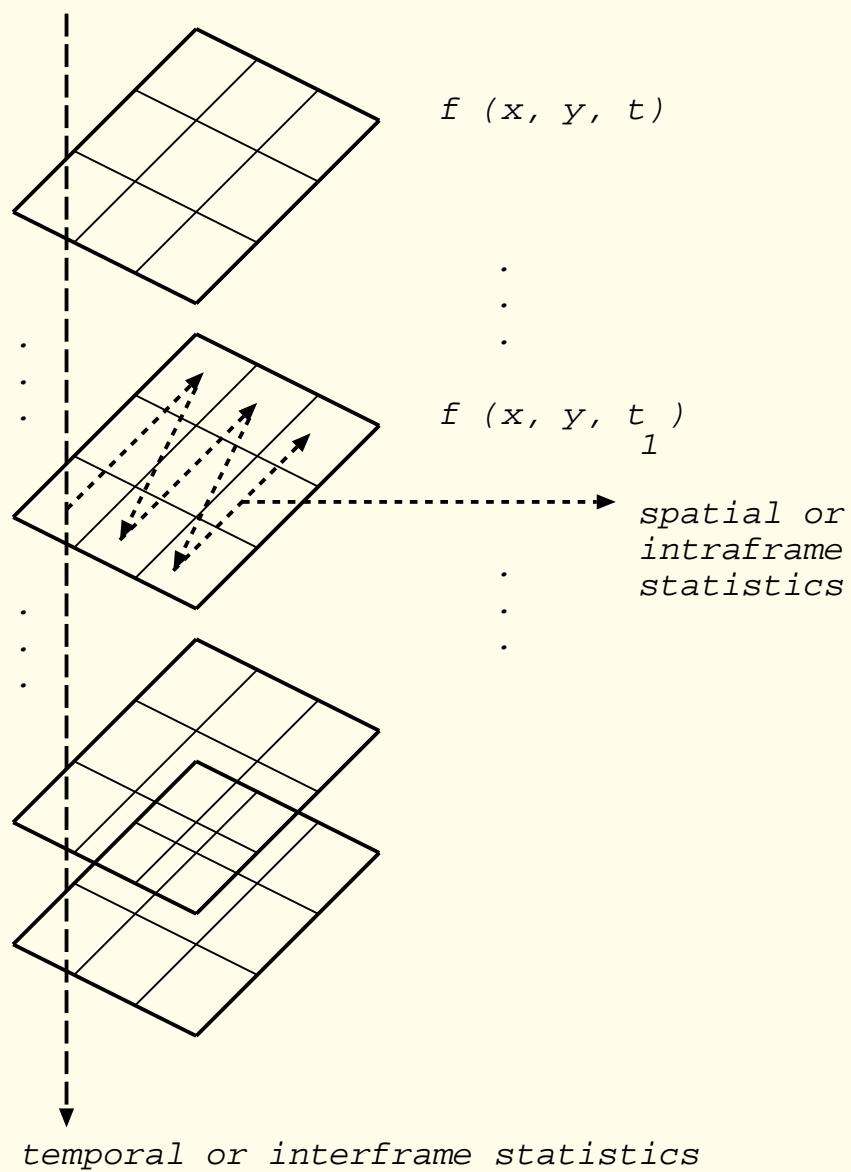


Figure 3.5: Spatial and temporal statistics of a video signal.



The mean of an image represents its average intensity or density; removal of the mean leads to an image with only the edges.

ACF: important role in the characterization of random processes.

Fourier transform [ACF] = power spectral density (PSD):
useful in frequency-domain analysis.



Statistical functions as above are useful in

- analysis of the behavior of random processes,
- modeling,
- spectrum analysis,
- filter design,
- data compression, and
- data communication.



3.1.2 Examples of noise PDFs

Gaussian:

$$p_x(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left[-\frac{(x - \mu_x)^2}{2\sigma_x^2}\right]. \quad (3.14)$$

Completely specified by mean μ_x and variance σ_x^2 .

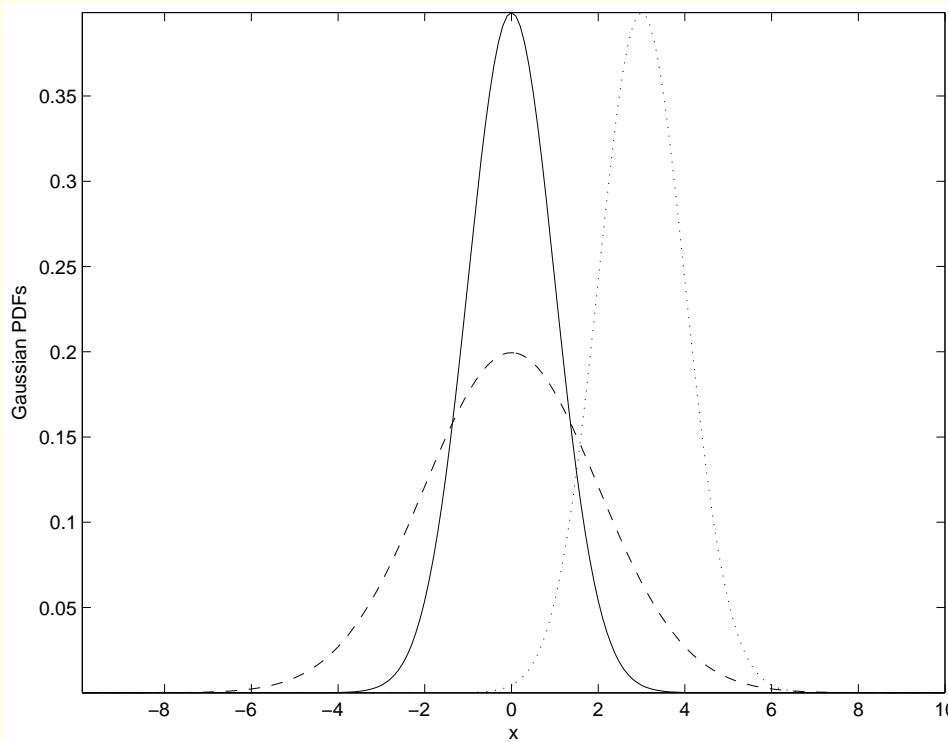


Figure 3.6: Three Gaussian PDFs. Solid line: $\mu = 0, \sigma = 1$. Dashed line: $\mu = 0, \sigma = 2$. Dotted line: $\mu = 3, \sigma = 1$.



When we have two jointly normal random processes x and y , the bivariate normal PDF is given by

$$p_{x,y}(x, y) = \frac{1}{\sqrt{4\pi^2(1 - \gamma^2)} \sigma_x \sigma_y} \times \quad (3.15)$$

$$\exp \left\{ -\frac{1}{2(1 - \gamma^2)} \left[\frac{(x - \mu_x)^2}{\sigma_x^2} - \frac{2\gamma(x - \mu_x)(y - \mu_y)}{\sigma_x \sigma_y} + \frac{(y - \mu_y)^2}{\sigma_y^2} \right] \right\}.$$



γ is the correlation coefficient given by

$$\gamma = \frac{E[(x - \mu_x)(y - \mu_y)]}{\sigma_x \sigma_y}. \quad (3.16)$$

If $\gamma = 0$, the two processes are uncorrelated.

The bivariate normal PDF then reduces to a product of two univariate Gaussians, which implies that the two processes are statistically independent.



Central limit theorem:

The PDF of a random process that is the sum of several statistically independent random processes is equal to the cascaded convolution of their individual PDFs.

When a large number of functions are convolved in cascade, the result tends toward a Gaussian-shaped function

regardless of the forms of the individual functions.



Uniform:

All possible values of a uniformly distributed random process have equal probability of occurrence.

PDF: a rectangle of height $\frac{1}{(b-a)}$ over the range (a, b) .

$$\mu_x = \frac{(a + b)}{2}.$$

$$\sigma_x^2 = \frac{(b - a)^2}{12}.$$

The quantization of gray levels in an image to a finite number of integers leads to an error or noise that is uniformly distributed.

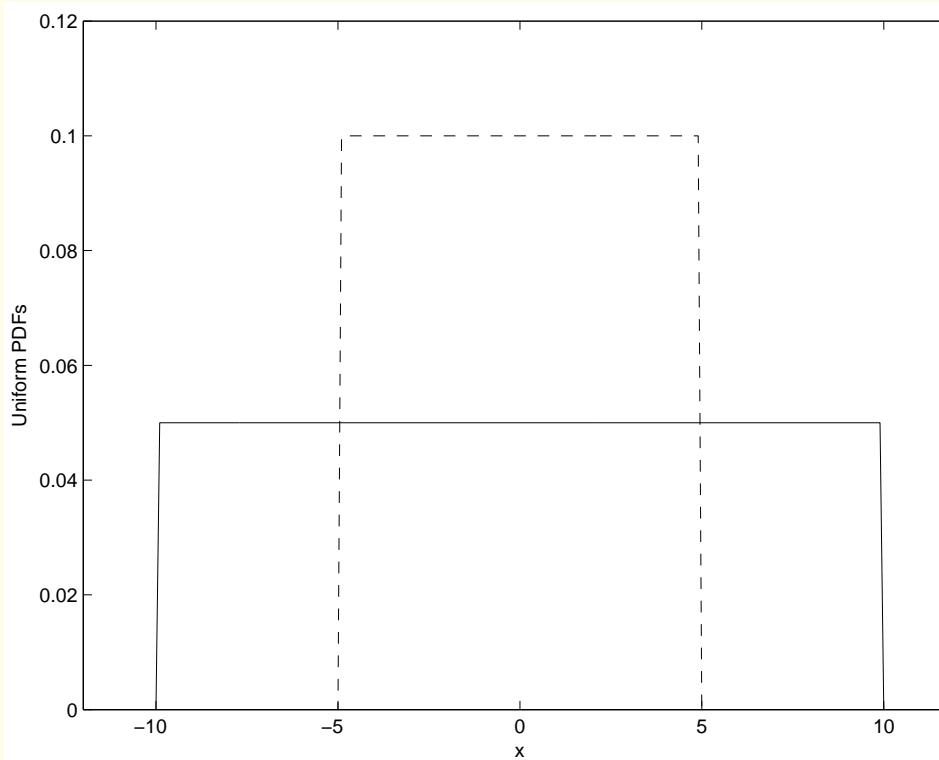


Figure 3.7: Two uniform PDFs. Solid line: $\mu = 0$, range = $(-10, 10)$. Dashed line: $\mu = 0$, range = $(-5, 5)$.



Poisson:

The counting of discrete random events such as the number of photons emitted by a source or detected by a sensor in a given interval of time leads to a random variable with a Poisson PDF.

The discrete nature of photons and the statistical randomness in their emission and detection contribute to uncertainty:

quantum noise, photon noise, mottle, Poisson noise in images.

Shot noise in electronic devices also modeled as Poisson noise.



One of the formulations of the Poisson PDF:

Probability that k photons are detected in a certain interval

$$P(k) = \exp(-\mu) \frac{\mu^k}{k!}. \quad (3.17)$$

μ = mean of the process = average number of photons counted in the specified interval over many trials.

Values of $P(k)$ for all (integer) k is the Poisson PDF.

Variance of the Poisson PDF = its mean.

Poisson PDF tends toward the Gaussian PDF for large mean.

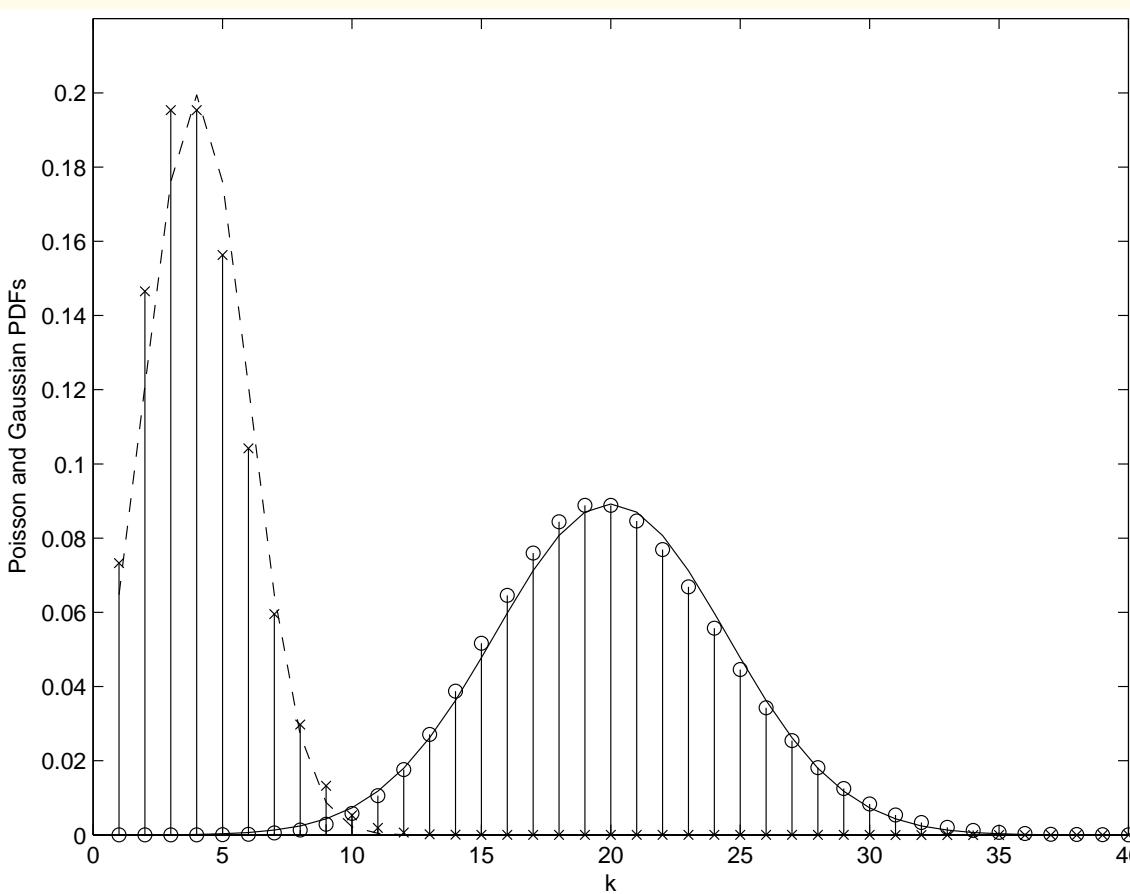


Figure 3.8: Two Poisson PDFs with the corresponding Gaussian PDFs superimposed. Bars with \times and dashed envelope: $\mu = \sigma^2 = 4$. Bars with \circ and solid envelope: $\mu = \sigma^2 = 20$.



Laplacian:

$$p_x(x) = \frac{1}{\sqrt{2} \sigma_x} \exp \left\{ - \frac{\sqrt{2} |x - \mu_x|}{\sigma_x} \right\}, \quad (3.18)$$

μ_x and σ_x^2 are the mean and variance.

Error values in linear prediction have Laplacian PDFs.

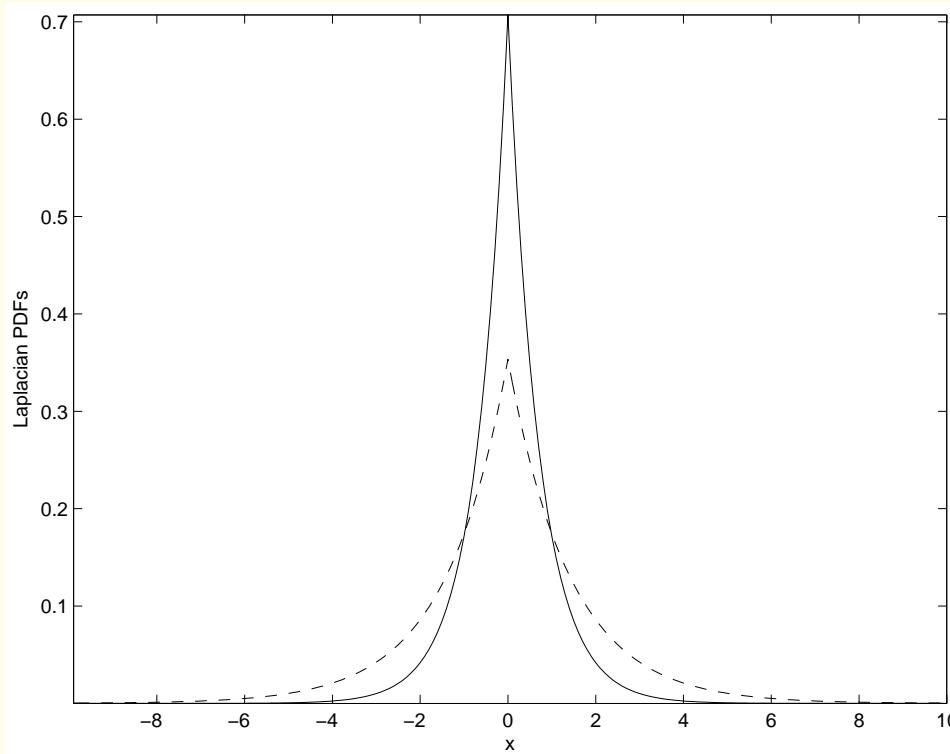


Figure 3.9: Two Laplacian PDFs with $\mu = 0$, $\sigma^2 = 1$ (solid) and $\mu = 0$, $\sigma^2 = 4$ (dashed).



Rayleigh:

$$p_x(x) = \frac{2}{b} (x - a) \exp\left\{-\frac{(x - a)^2}{b}\right\} u(x - a), \quad (3.19)$$

where $u(x)$ is the unit step function

$$u(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3.20)$$



Mean and variance of the Rayleigh PDF determined by a and b :

$$\mu_x = a + \sqrt{(\pi b / 4)}.$$

$$\sigma_x^2 = b(4 - \pi) / 4.$$

Rayleigh PDF used to model speckle noise.

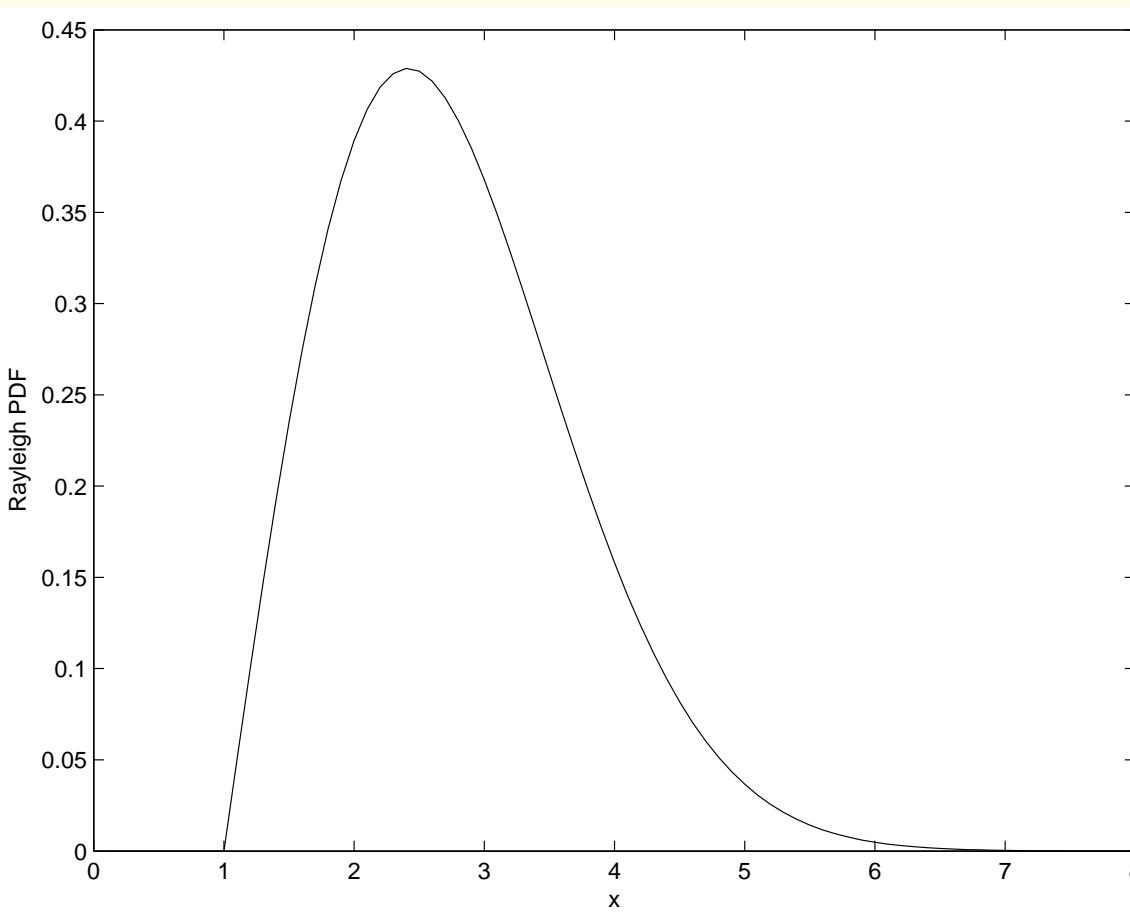


Figure 3.10: Rayleigh PDF with $a = 1$ and $b = 4$.



3.1.3 Structured noise

Structured noise in biomedical images:

- X-ray grid artifact.
- Grid or frame used for image registration and calibration.
- Labels indicating patient identification, patient positioning, imaging parameters.
- Surgical implants such as staples, pins, and screws
- Power-line interference at 50 Hz or 60 Hz is common in biomedical signals, but not common in biomedical images.



3.1.4 *Physiological interference*

A patient or experimental subject may not be able to exercise control on all of his or her physiological processes and systems.

- Effect of breathing on a chest X-ray image.
- Effect of breathing, peristalsis, and movement of material through the gastro-intestinal system.
- Effect of cardiovascular activity on CT images of the chest.
- Effect of pulsatile movement of arteries in DSA.

Anatomical details such as the ribs in chest X-ray images and the skull in brain imaging may be considered to be artifacts when other details are of primary interest.



3.1.5 Other types of noise and artifact

Systematic errors caused by

- geometric distortion,
- miscalibration,
- nonlinear response of detectors,
- sampling, and
- quantization.



A few other types of artifact:

- Punctate or shot noise due to dust on the screen, film, or examination table.
- Scratches on film that could appear as intense line segments.
- Shot noise due to inactive elements in a detector array.
- Salt-and-pepper noise due to impulsive noise.
- Film-grain noise due to scanning of films with high resolution.
- Punctate noise in chest X-ray or mammographic images caused by cosmetic powder or deodorant (calcifications?!).
- Superimposed images of clothing accessories such as pins, hooks, buttons, and jewelry.



3.1.6 Stationary versus nonstationary processes

Random processes may be characterized in terms of their temporal/spatial and/or ensemble statistics.

A random process is said to be *stationary in the strict sense* or *strongly stationary* if its statistics are not affected by a shift in the origin of time or space.

In most practical applications, only the first-order and second-order averages are used.



A random process is said to be

weakly stationary or *stationary in the wide sense*

if its mean is a constant and its ACF depends only upon the difference (or shift) in time or space:

$$\mu_f(x_1, y_1) = \mu_f \text{ and } \phi_f(x_1, x_1 + \alpha, y_1, y_1 + \beta) = \phi_f(\alpha, \beta).$$

The ACF is now a function of the shift parameters α and β only; the PSD of the process does not vary with space.



A stationary process is said to be *ergodic* if the temporal statistics computed are independent of the sample observed; same results obtained with any sample observation $f_k(t)$.

The time averages of the process are then independent of k :
 $\mu_f(k) = \mu_f$ and $\phi_f(\tau, k) = \phi_f(\tau)$.

All ensemble statistics may be replaced by temporal statistics when analyzing ergodic processes.

Ergodic processes are an important type of stationary random processes because their statistics may be computed from a single observation as a function of time.

The concept may be extended to functions of space, but the term “ergodic” is not commonly applied to images.



Nonstationary processes: statistics vary with time or space.

The statistics of most images vary over space;
such variations are the source of pictorial information.

Most biomedical systems are dynamic systems and produce
nonstationary signals and images.

A physical or physiological system has limitations in the rate at
which it can change its characteristics.

This facilitates the breaking of a signal into segments of short
duration (a few milliseconds), over which its statistics may be
assumed to remain constant: *quasistationary process*.



Techniques designed for stationary signals may then be extended and applied to nonstationary signals.

Analysis of signals by this approach: *short-time analysis*.

The characteristics of the features in an image vary over relatively large scales of space.

Statistical parameters within small regions of space, an object, or an organ may be assumed to remain constant.

The image may then be assumed to be *block-wise stationary*:

sectioned, block-by-block, or moving-window processing techniques may be applied.

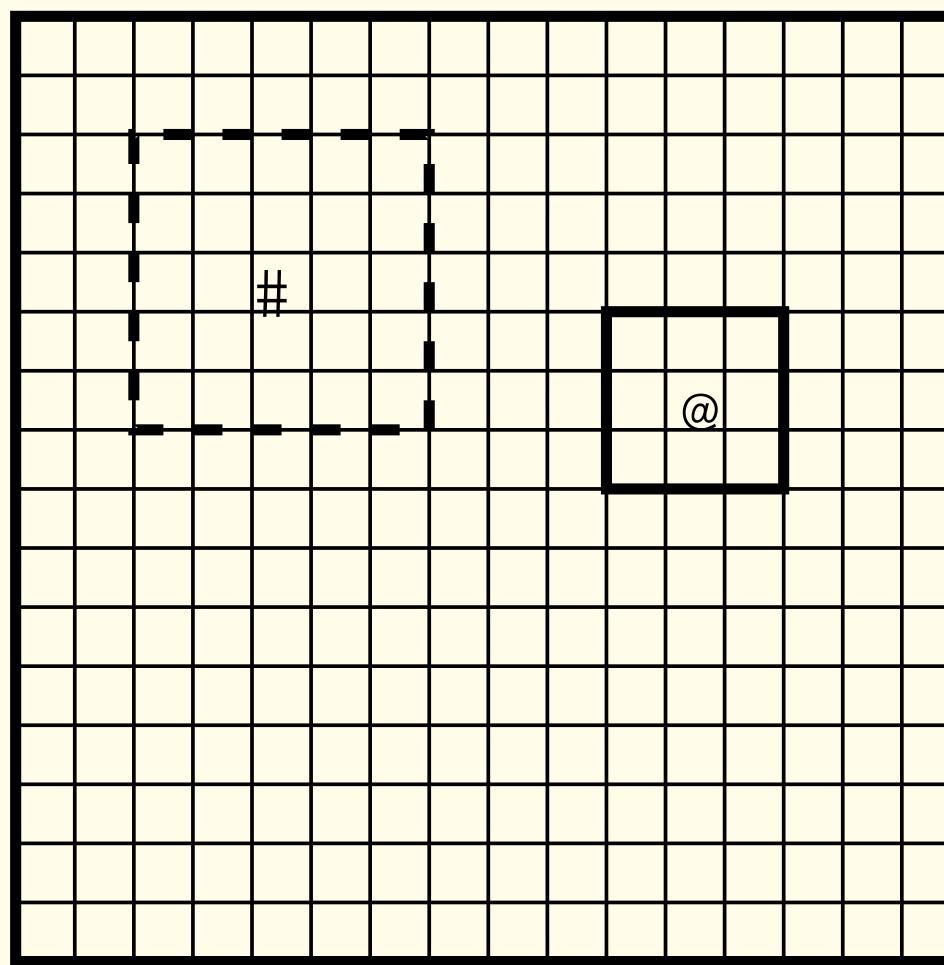


Figure 3.11: Block-by-block processing of an image. Statistics computed by using the pixels within the window shown with solid lines (3×3 pixels) are applicable to the pixel marked with the @ symbol. Statistics for use when processing the pixel marked with the # symbol (5×5 pixels) are computed by using the pixels within the window shown with dashed lines.



The cardiac system performs rhythmic operations:
Such a system is nonstationary.

Various phases of the cardiac cycle — as well as the related components of the associated electrocardiogram (ECG) and phonocardiogram (PCG) — repeat in an almost-periodic manner:

cyclo-stationary signals.

The cyclical repetition of the process facilitates synchronized ensemble averaging using epochs or events extracted from an observation of the signal over many cycles.



In gated blood-pool imaging, nuclear medicine images of the heart are acquired in several parts over short intervals of time.

Images acquired at the same phases of the cardiac cycle — determined by using the ECG signal as a reference, trigger, or “gating” signal — are accumulated over several cardiac cycles.

A sequence of such gated and averaged frames over a full cardiac cycle may then be played as a video or a movie.



3.1.7 Covariance and cross-correlation

When two random processes f and g need to be compared, we could compute the covariance:

$$\sigma_{fg} = E[(f - \mu_f)(g - \mu_g)] \quad (3.21)$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (f - \mu_f)(g - \mu_g) p_{f,g}(f, g) df dg,$$

$p_{f,g}(f, g)$ is the joint PDF of the two processes.

Image coordinates omitted for compact notation.



Covariance normalized to get correlation coefficient:

$$\rho_{fg} = \frac{\sigma_{fg}}{\sigma_f \sigma_g}, \quad (3.22)$$

$$-1 \leq \rho_{fg} \leq +1.$$

High covariance indicates that the two processes have similar statistical variability or behavior.

The processes f and g are uncorrelated if $\rho_{fg} = 0$.

Statistically independent processes are also uncorrelated;

the converse of this property is, in general, not true.



With random processes f and g that are functions of space, the cross-correlation function (CCF) between them is

$$\phi_{fg}(\alpha, \beta) = E[f(x, y) g(x + \alpha, y + \beta)]. \quad (3.23)$$

Correlation functions are useful in analyzing the nature of variability and spectral bandwidth of images,

and for the detection of objects by *template matching*.



3.1.8 *Signal-dependent noise*

Noise may be categorized as being independent of the signal of interest if no statistical parameter of any order of the noise process is a function of the signal.

Several cases exist in biomedical imaging where this assumption is not valid:

the noise is functionally related to or dependent upon the signal.



Poisson noise:

Imaging systems that operate in low-light conditions, or in low-dose radiation conditions such as nuclear medicine imaging, are often affected by photon noise.

Probabilistic description of an observed image (pixel) $g_o(m, n)$:

$$P(g_o(m, n)|f(m, n), \lambda) = \quad (3.24)$$

$$\frac{[\lambda f(m, n)]^{g_o(m,n)} \exp[-\lambda f(m, n)]}{g_o(m, n)!}.$$



$f(m, n)$ is the undegraded pixel value,

λ is a proportionality factor.

$$E[g_o(m, n)] = \lambda E[f(m, n)]. \quad (3.25)$$

Images corrupted with Poisson noise are usually normalized as

$$g(m, n) = \frac{g_o(m, n)}{\lambda}. \quad (3.26)$$



Poisson noise may be modeled as stationary, additive noise uncorrelated with the signal, with zero mean and variance

$$\sigma_{\eta}^2(m, n) = \frac{E[f(m, n)]}{\lambda} = \frac{E[g(m, n)]}{\lambda}. \quad (3.27)$$



Film-grain noise:

The granular structure of film due to the silver-halide grains contributes noise.

Model of Froehlich et al. for image with film-grain noise:

$$g(m, n) = f(m, n) + \kappa \mathcal{F}[f(m, n)] \eta_1(m, n) + \eta_2(m, n). \quad (3.28)$$

κ is a proportionality factor.



$\mathcal{F}[\cdot]$ is a mathematical function.

$\eta_1(m, n)$ and $\eta_2(m, n)$ are samples from two random processes independent of the signal.

Includes signal-independent as well as signal-dependent noise;

noise could be additive or multiplicative.

Reduces to signal-independent additive noise model if $\kappa = 0$.



Froehlich et al. modeled film-grain noise with

$$\mathcal{F}[f(m, n)] = [f(m, n)]^p, \text{ using } p = 0.5.$$

η_1 and η_2 were assumed to be Gaussian-distributed, uncorrelated, zero-mean random processes:

one component signal-dependent through the factor

$$\kappa \sqrt{f(m, n)} \eta_1(m, n),$$

another that is signal-independent: $\eta_2(m, n)$.



Film-grain noise may be modeled as additive noise with

$$\eta(m, n) = \kappa \sqrt{f(m, n)} \eta_1(m, n) + \eta_2(m, n).$$

$\eta(m, n)$ as above is stationary, with zero mean and variance

$$\sigma_\eta^2(m, n) = \kappa^2 E[g(m, n)] \sigma_{\eta_1}^2 + \sigma_{\eta_2}^2. \quad (3.29)$$



Speckle noise:

Corrupts images obtained by coherent radiation, such as synthetic-aperture radar (SAR), ultrasound, laser, and sonar.

Multiplicative model:

$$g(m, n) = f(m, n) \eta_1(m, n), \quad (3.30)$$

$\eta_1(m, n)$ is a stationary process uncorrelated with the image.

If $\mu_{\eta_1} \neq 1$, the noisy image normalized by dividing by μ_{η_1} .



Depending upon the application, the distribution of the noise may be exponential, Gaussian, or Rayleigh.

The multiplicative model may be converted to an additive model with $\eta(m, n)$ being zero-mean additive noise having a space-variant, signal-dependent variance given by

$$\sigma_{\eta}^2(m, n) = \frac{\sigma_{\eta_1}^2}{1 + \sigma_{\eta_1}^2} [\sigma_g^2(m, n) + \mu_g^2(m, n)]. \quad (3.31)$$

$\sigma_g^2(m, n)$ and $\mu_g(m, n)$ are the variance and the mean of the noisy image at the point (m, n) , respectively.



Transformation of signal-dependent noise to signal-independent noise:

In the model used by Naderi & Sawchuk and Arsenault et al., the signal-independent component of the noise as in Equation 3.28 is assumed to be zero.

By applying an appropriate transformation to the whole image, the noise can be made signal-independent; e.g.,

$$T[g(m, n)] = \alpha \sqrt{g(m, n)}, \quad (3.32)$$

where α is an appropriate normalizing constant.

The noise in the transformed image is additive, has a Gaussian distribution, is unbiased, and has a standard deviation that no longer depends on the signal but is given by $\frac{\alpha \kappa}{2}$.



3.2 Synchronized or Multiframe Averaging

If the object can remain free from motion (internal or external) over a long period of time compared to the time required to record an image,

it becomes possible to acquire several frames of images of the object in precisely the same state or condition.

Then, the frames may be averaged to reduce noise;

this is known as *multiframe averaging*.



The method may be extended to the imaging of dynamic systems whose movements follow a rhythm or cycle with phases that can be determined by another signal,

such as the cardiac system whose phases of contraction are indicated by the ECG signal.

Then, several image frames may be acquired at the same phase of the rhythmic movement over successive cycles, and averaged.

Such a process is known as *synchronized averaging*.

Averaging as above is a form of *ensemble averaging*.



Let us represent a single image frame as

$$g_i(x, y) = f(x, y) + \eta_i(x, y), \quad (3.33)$$

$g_i(x, y)$ is the i^{th} observed frame of $f(x, y)$,

$\eta_i(x, y)$ is the noise in the same frame.

Assume that the noise is independent of the signal.

Original image $f(x, y)$ is invariant from one frame to another.

It follows that $\sigma_{g_i(x, y)}^2 = \sigma_{\eta_i(x, y)}^2$.



If M frames of the image are acquired, the averaged image

$$\bar{g}(x, y) = \frac{1}{M} \sum_{i=1}^M g_i(x, y). \quad (3.34)$$

If the mean of the noise process is zero,

$$\sum_{i=1}^M \eta_i(x, y) \rightarrow 0 \text{ as } M \rightarrow \infty.$$



$$E[\bar{g}(x, y)] = f(x, y). \quad (3.35)$$

$$\sigma_{\bar{g}(x, y)}^2 = \frac{1}{M} \sigma_{\eta(x, y)}^2. \quad (3.36)$$

The variance is reduced by a factor of $\frac{1}{M}$;

the SNR is improved by the factor \sqrt{M} .

Motion or change between the frames will lead to distortion.

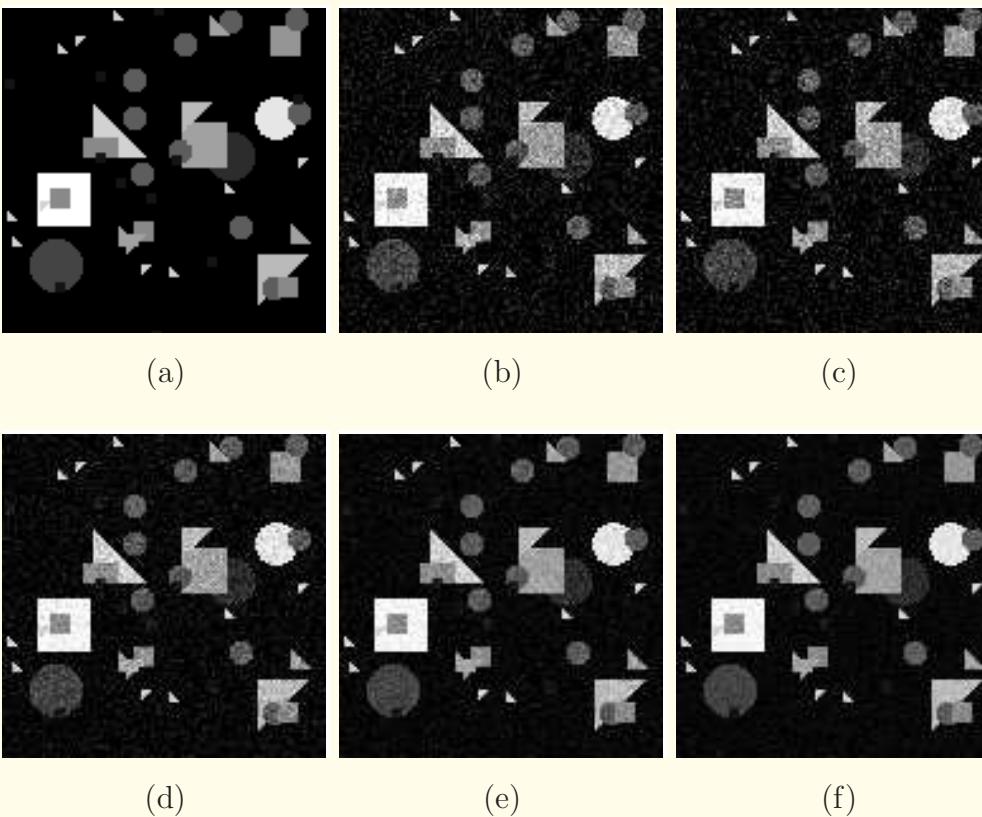


Figure 3.12: (a) “Shapes”: a 128×128 test image with various geometrical objects placed at random. (b) Image in (a) with Gaussian noise added, with $\mu = 0$, $\sigma^2 = 0.01$ (normalized), RMS error = 19.32. (c) Second version of noisy image, RMS error = 19.54. Result of multiframe averaging using (d) the two frames in (b) and (c), RMS error = 15.30; (e) four frames, RMS error = 12.51; (f) eight frames, RMS error = 10.99.



3.3 Space-domain Local-statistics-based Filters

Given a single, noisy observation of an image of finite size.

We do not have access to an ensemble of images to perform multiframe (synchronized) averaging.

Spatial statistics computed over the entire image frame will lead to scalar values that do not assist in removing the noise.

We should also accommodate for nonstationarity of the image.



Moving-window filtering using windows of small size such as 3×3 , 5×5 , or 7×7 pixels becomes a valuable option.

Rectangular windows as well as windows of other shapes may also be considered where appropriate.



Various statistics of the pixels within moving window computed.

Result applied to the pixel in the output at the location where the window is placed (centered) on the input image.

Only the pixel values in the input image are used.

The output is stored in a separate array.

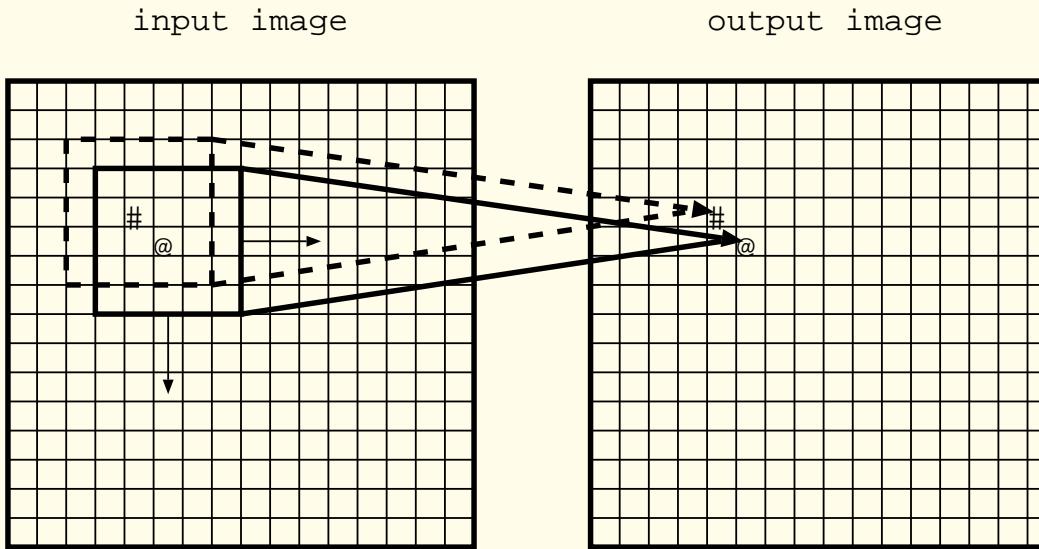
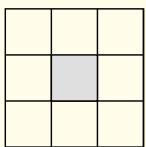
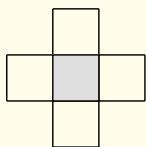
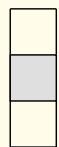
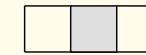


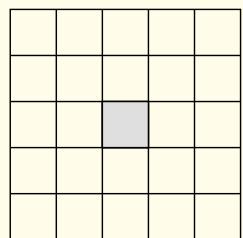
Figure 3.13: Moving-window filtering of an image. The size of the moving window in the illustration is 5×5 pixels. Statistics computed by using the pixels within the window are applied to the pixel at the same location in the output image. The moving window is shown for two pixel locations marked # and @.

(a) 3x3 square
(8-connected)(b) 4-connected
or integer
distance 1

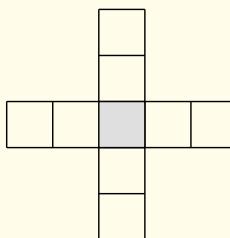
(c) 3x1 bar



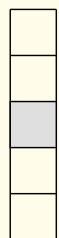
(d) 1x3 bar



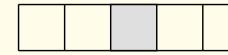
(e) 5x5 square



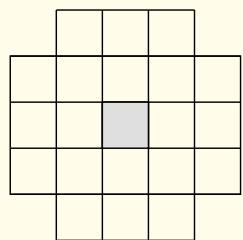
(f) cross



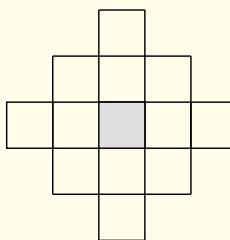
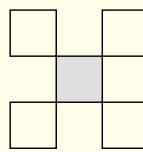
(g) 5x1 bar



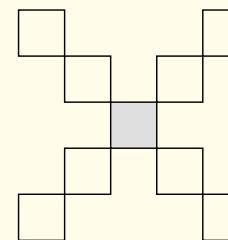
(h) 1x5 bar



(i) circle

(j) integer
distance 2

(k) X-1



(l) X-2

Figure 3.14: A few commonly used moving-window neighborhood shapes for image filtering. The result computed by using the pixels within a window is applied to the pixel at the location of its center, shown shaded, in the output image.



3.3.1 *The mean filter*

If we were to select the pixels in a small neighborhood around the pixel to be processed, the following assumptions may be made:

- image component is relatively constant (quasistationary);
- the only variations in the neighborhood are due to noise;
- noise is additive,
- noise is independent of the image, and
- noise has zero mean.



If we take the mean of the pixels in the neighborhood, the result will tend toward the true pixel value in the original image.

In essence, a spatial collection of pixels around the pixel being processed is substituted for an ensemble of pixels at the same location from multiple frames in the averaging process:

the image-generating process is assumed to be *ergodic*.



3×3 or 8-connected neighborhood for mean filtering:

$$g(m, n) = \frac{1}{9} \sum_{\alpha=-1}^1 \sum_{\beta=-1}^1 f(m + \alpha, n + \beta). \quad (3.37)$$

$$\begin{aligned} g(m, n) = & \frac{1}{9} \times \\ & [f(m - 1, n - 1) + f(m - 1, n) + f(m - 1, n + 1) \\ & + f(m, n - 1) + f(m, n) + f(m, n + 1) \\ & + f(m + 1, n - 1) + f(m + 1, n) + f(m + 1, n + 1)]. \end{aligned} \quad (3.38)$$



This is equivalent to convolution with the 3×3 *convolution mask*

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (3.39)$$



Filter as above cannot be directly applied at the edges of the input image array:

- (a) extend the input array with a border of zero-valued pixels;
- (b) do not process the pixels at the edges;
- (c) replace edge pixels with average of available neighbors.



Mean filter can suppress Gaussian and uniformly distributed noise effectively in relatively homogeneous areas of an image.

The operation leads to blurring at the edges of the objects in the image, and loss of fine details and texture.

Blurring of edges minimized by not applying the mean filter if the difference between the pixel being processed and the mean of its neighbors is greater than a threshold (nonlinearity).



3.3.2 *The median filter*

The median of a collection of samples is the value that splits the population in half:

half the number of pixels will have values less than the median and half will have values greater than the median.

The most common procedure rank-orders the pixels in a neighborhood containing an odd number of pixels, and the pixel value at the middle of the list is selected as the median.

The median filter is a nonlinear filter.



Median filter provides better noise removal than the mean without blurring, especially when the noise has a long-tailed PDF (resulting in outliers) and in the case of salt-and-pepper noise.

Median filter could result in the clipping of corners and distortion of the shape of sharp-edged objects.

Median filtering with large neighborhoods could also result in the complete elimination of small objects.

Non-square neighborhoods are used for median filtering to limit clipping of corners and distortion of shape.

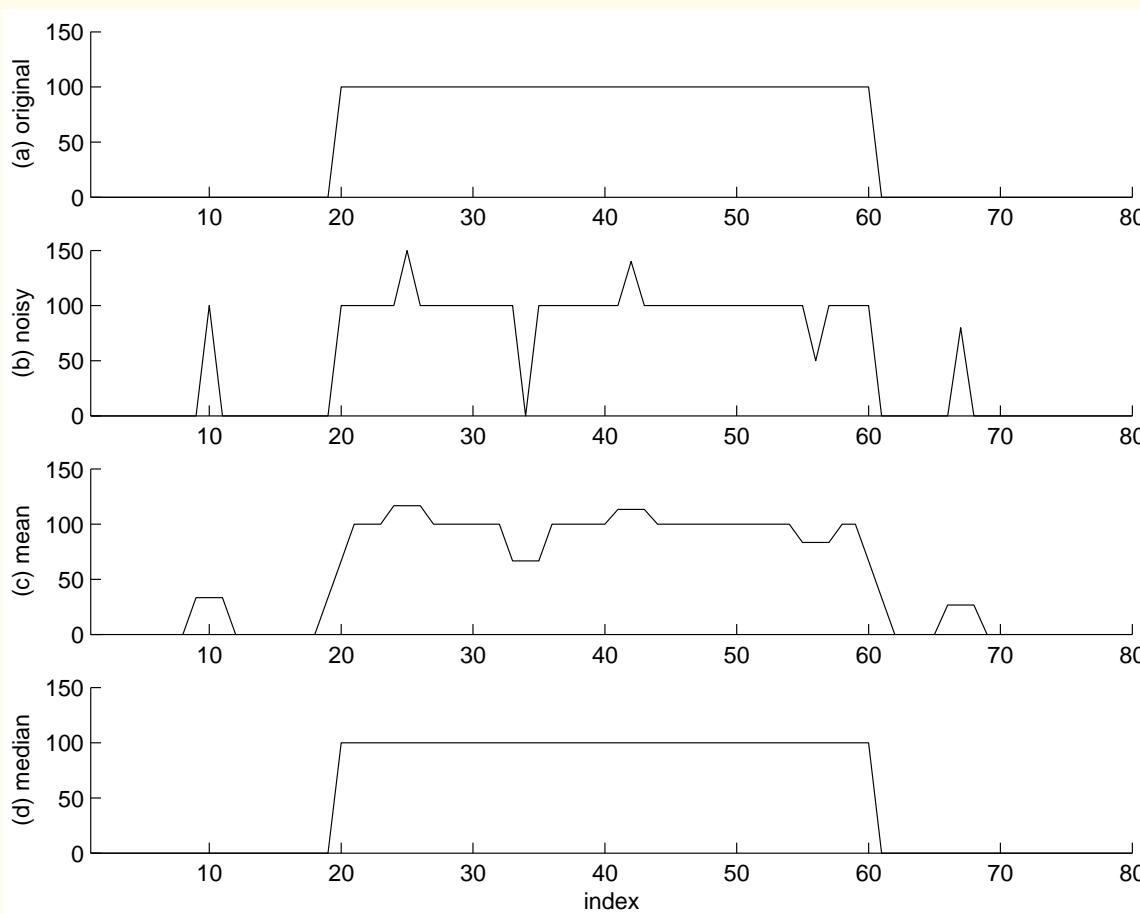


Figure 3.15: (a) A 1D test signal with a rectangular pulse. (b) Degraded signal with impulse or shot noise. Result of filtering the degraded signal using (c) the mean and (d) the median operation with a sliding window of $N = 3$ samples.

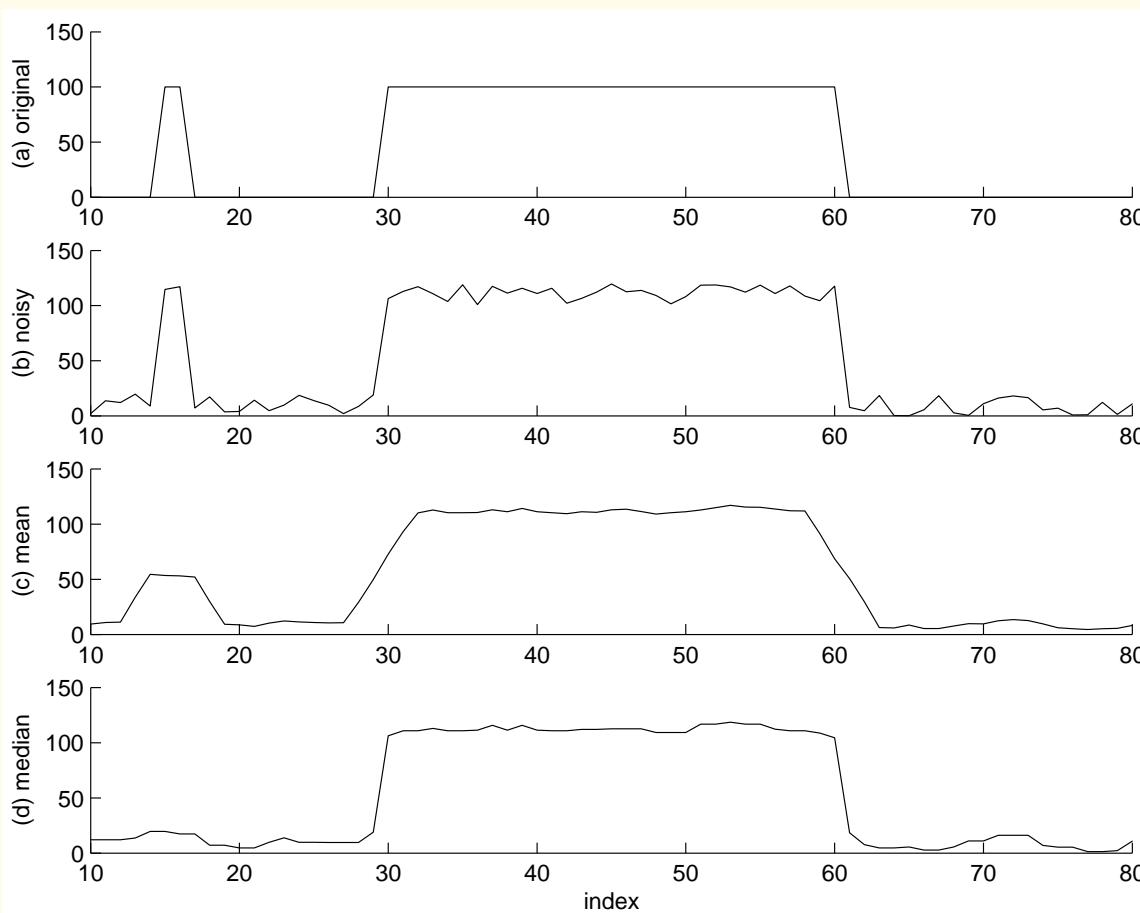


Figure 3.16: (a) A 1D test signal with two rectangular pulses. (b) Degraded signal with uniformly distributed noise. Result of filtering the degraded signal using (c) the mean, and (d) the median operation with a sliding window of $N = 5$ samples.



The RMS error (or the MSE) is commonly used to compare the results of various image processing operations.

Limitations exist in using the RMS error in comparing images with different types of artifact and distortion.

An image with a higher RMS error may present better visual quality than another image with a lower RMS error.

Visual inspection and analysis of the results by qualified users or experts in the domain of application is important.

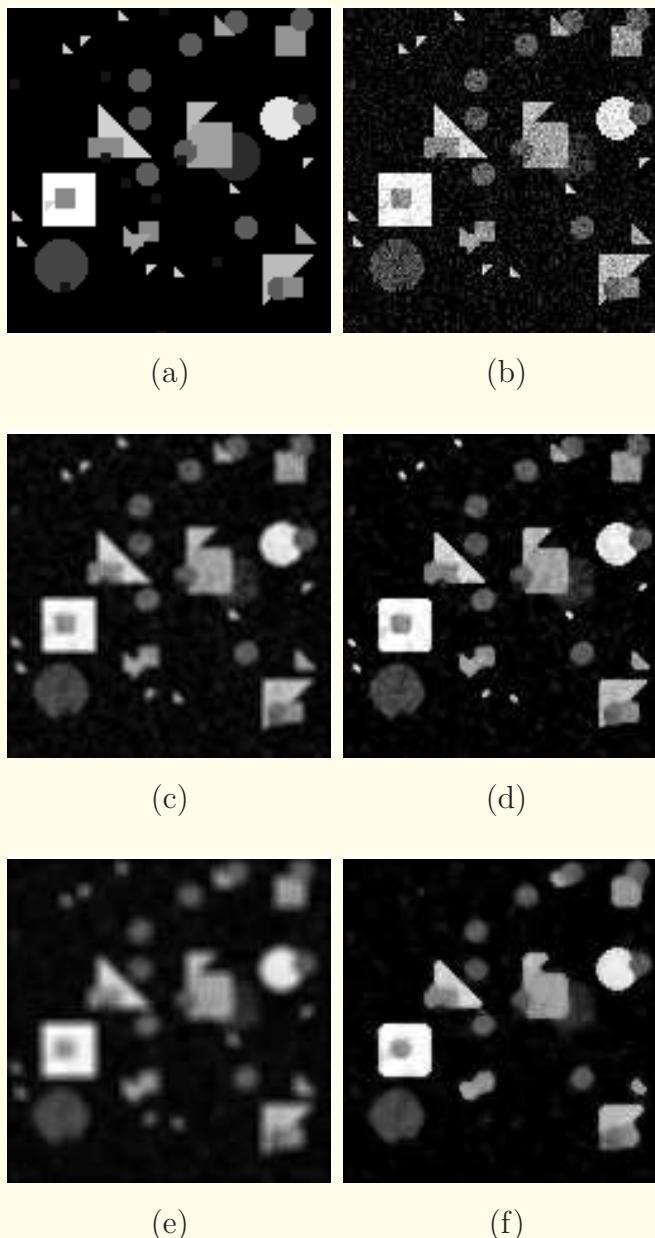


Figure 3.17: (a) Shapes test image. (b) Image in (a) with Gaussian noise added, with $\mu = 0$, $\sigma^2 = 0.01$ (normalized), RMS error = 19.56. Result of filtering the noisy image in (b) using: (c) 3×3 mean, RMS error = 22.62; (d) 3×3 median, RMS error = 15.40; (e) 5×5 mean, RMS error = 28.08; (f) 5×5 median, RMS error = 22.35.

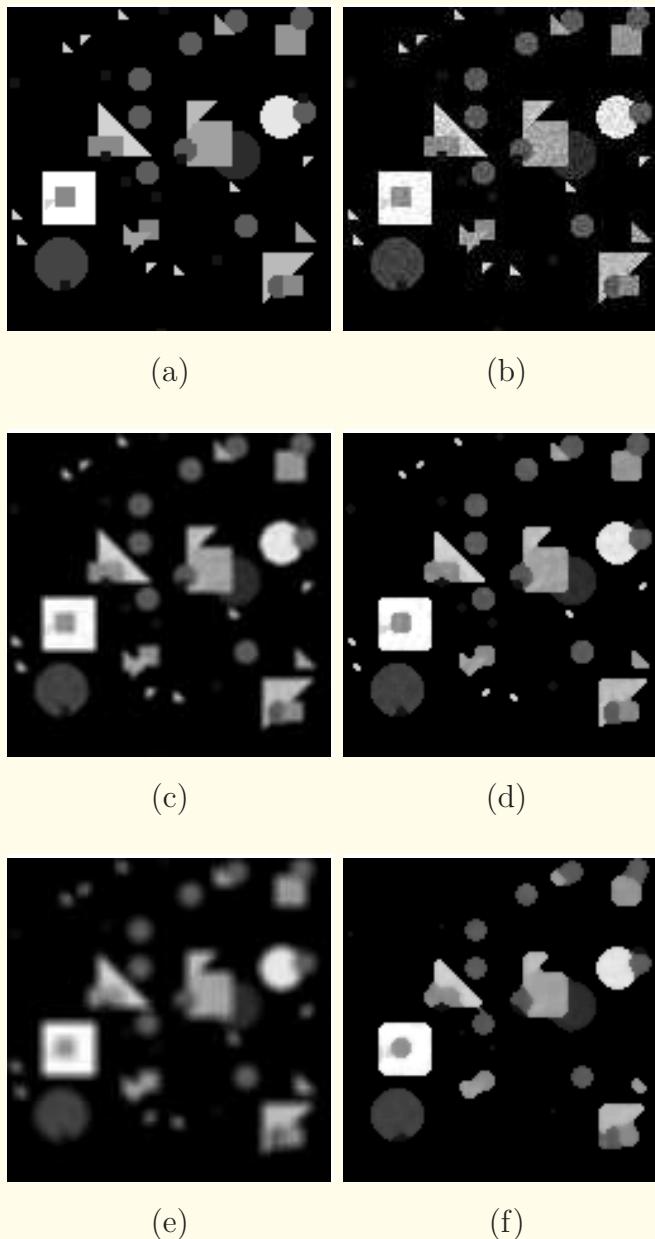


Figure 3.18: (a) Shapes test image. (b) Image in (a) with Poisson noise, RMS error = 5.00. Result of filtering the noisy image in (b) using: (c) 3×3 mean, RMS error = 19.40; (d) 3×3 median, RMS error = 13.19; (e) 5×5 mean, RMS error = 25.85; (f) 5×5 median, RMS error = 23.35.

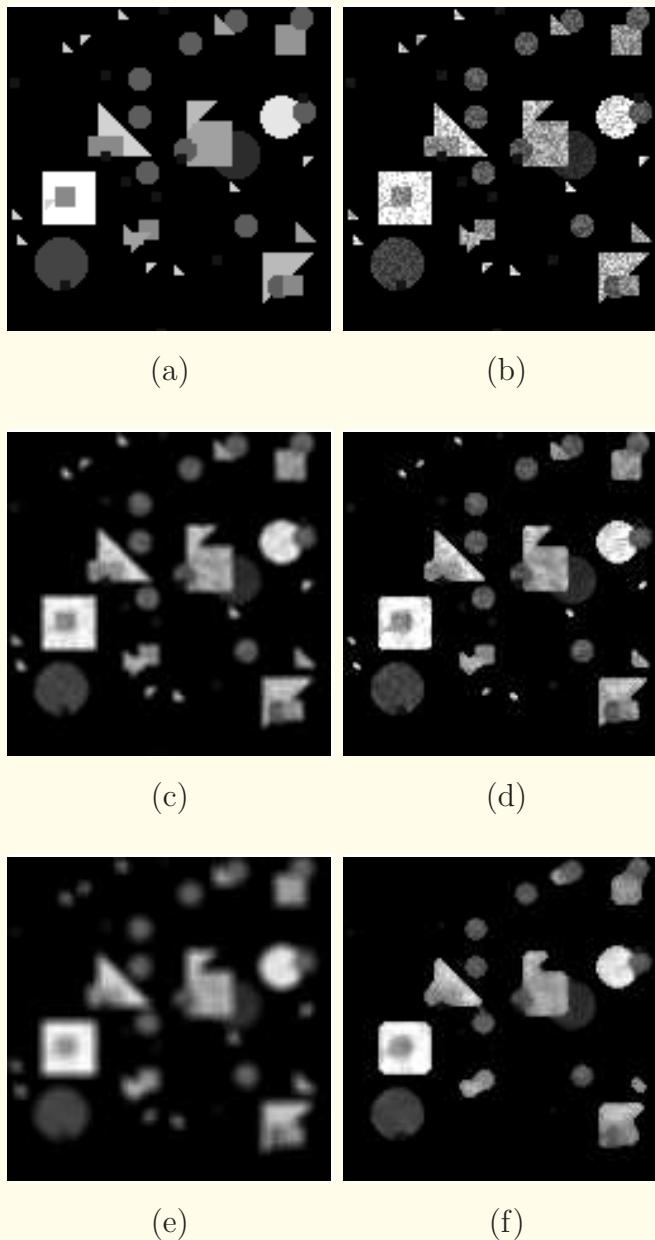


Figure 3.19: (a) Shapes test image. (b) Image in (a) with speckle noise, with $\mu = 0$, $\sigma^2 = 0.04$ (normalized), RMS error = 12.28. Result of filtering the noisy image in (b) using: (c) 3×3 mean, RMS error = 20.30; (d) 3×3 median, RMS error = 15.66; (e) 5×5 mean, RMS error = 26.32; (f) 5×5 median, RMS error = 24.56.

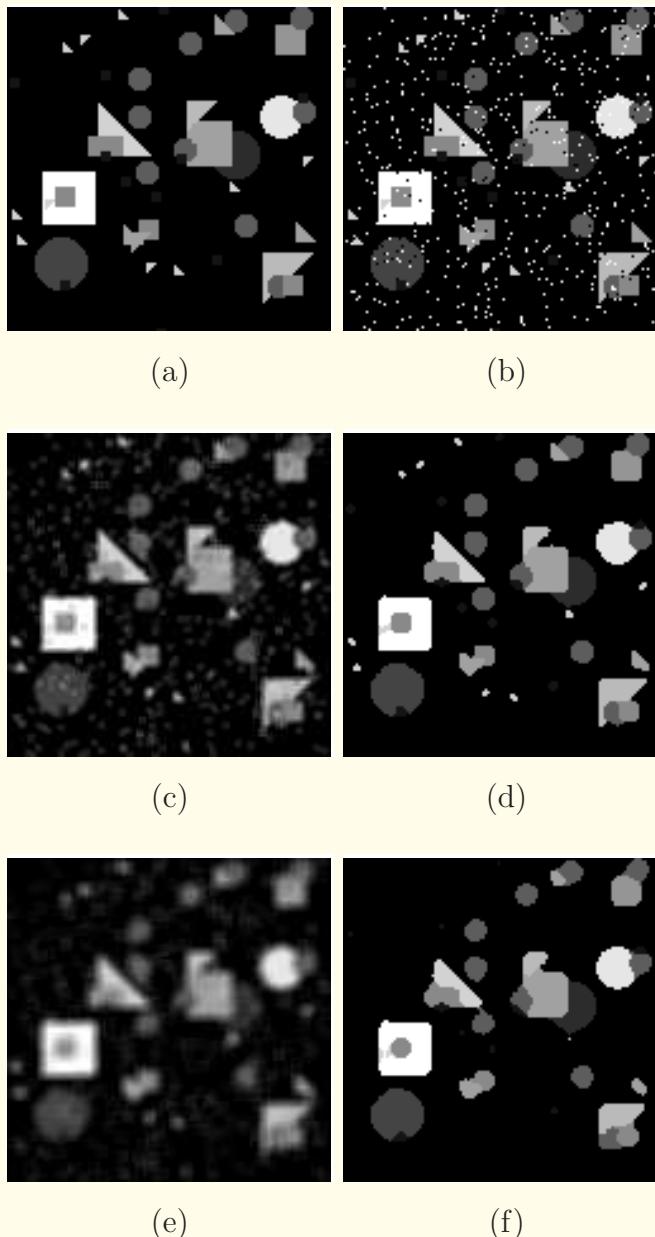


Figure 3.20: (a) Shapes test image. (b) Image in (a) with salt-and-pepper noise added, with density = 0.05, RMS error = 40.99. Result of filtering the noisy image in (b) using: (c) 3×3 mean, RMS error = 24.85; (d) 3×3 median, RMS error = 14.59; (e) 5×5 mean, RMS error = 28.24; (f) 5×5 median, RMS error = 23.14.

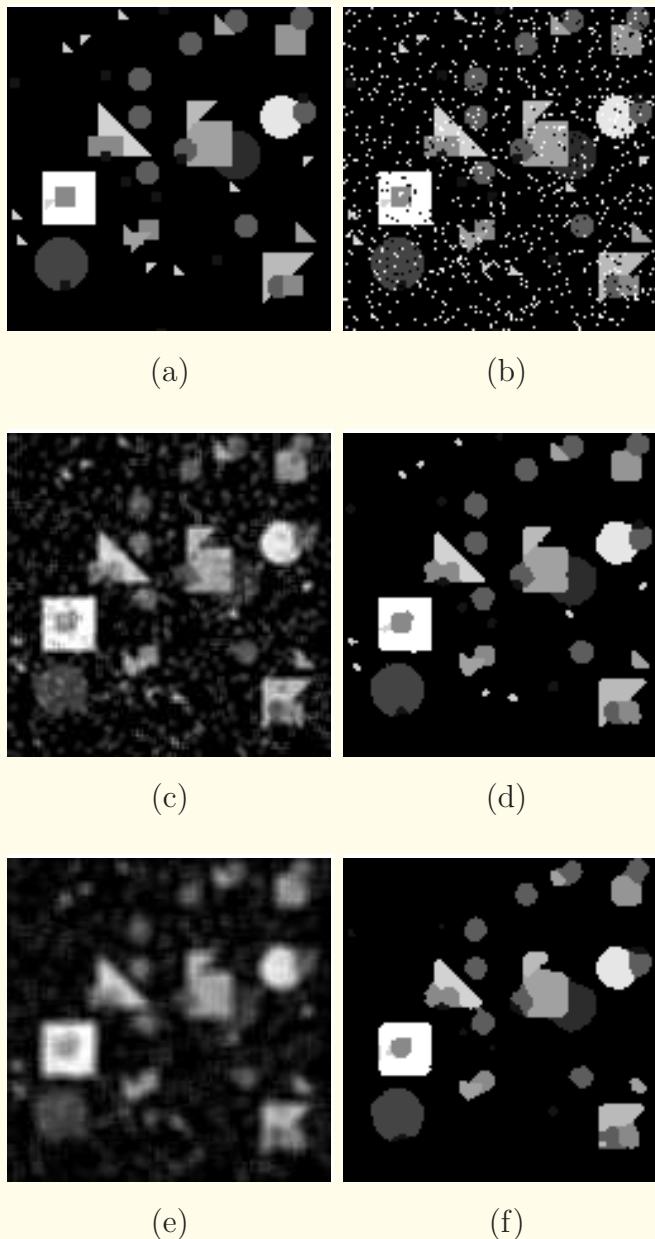


Figure 3.21: (a) Shapes test image. (b) Image in (a) with salt-and-pepper noise added, with density = 0.1, RMS error = 56.32. Result of filtering the noisy image in (b) using: (c) 3×3 mean, RMS error = 29.87; (d) 3×3 median, RMS error = 15.42; (e) 5×5 mean, RMS error = 31.25; (f) 5×5 median, RMS error = 23.32.

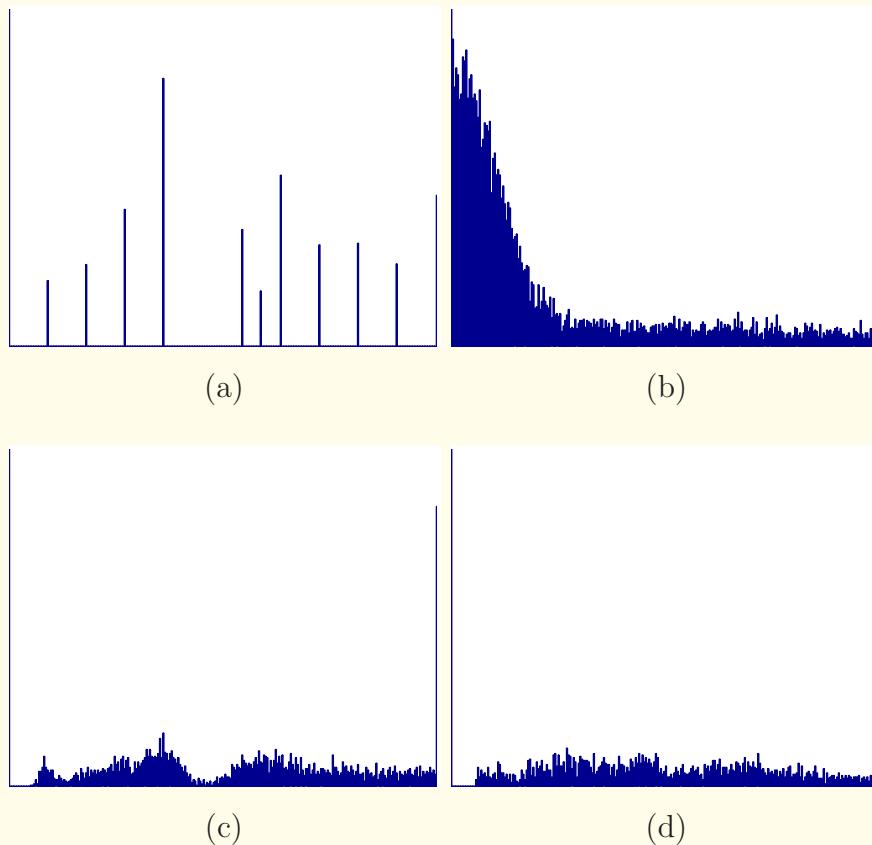


Figure 3.22: Normalized histograms of (a) the Shapes test image and of the image with (b) Gaussian noise, (c) Poisson noise, and (d) speckle noise. The first histogram has been scaled to display the range of probability $(0, 0.05)$ only; the remaining histograms have been scaled to display the range $(0, 0.015)$ only in order to show the important details. The probability values of gray levels 0 and 255 have been clipped in some of the histograms. Each histogram represents the gray-level range of $[0, 255]$.

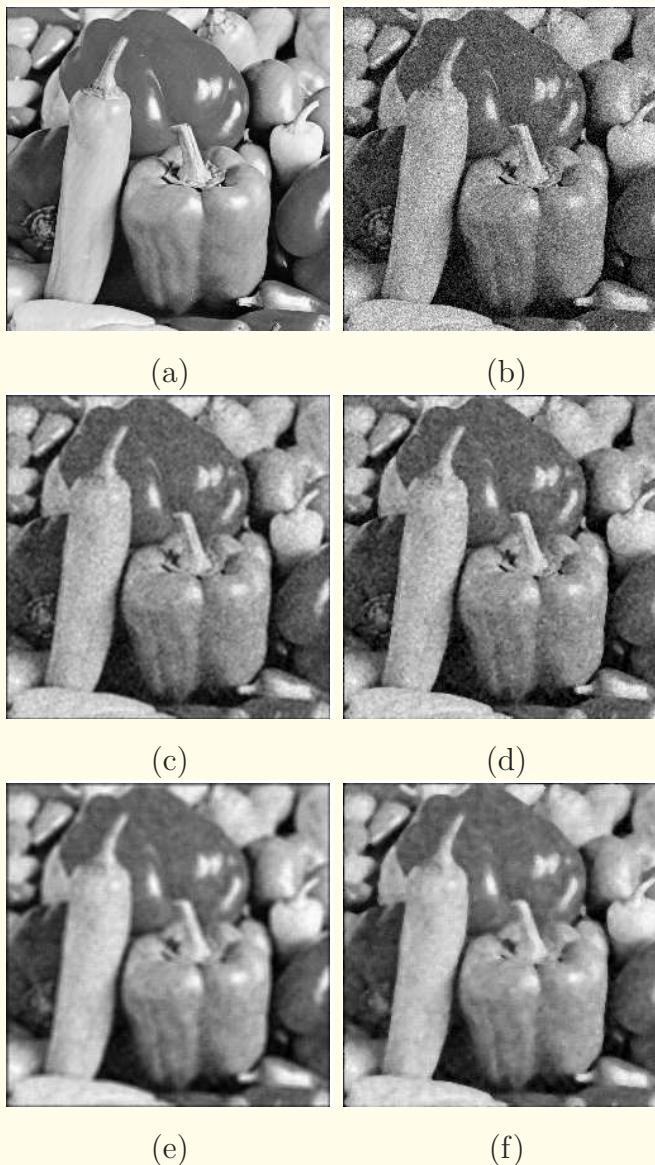


Figure 3.23: (a) “Peppers”: a 512×512 test image. (b) Image in (a) with Gaussian noise added, with $\mu = 0$, $\sigma^2 = 0.01$ (normalized), RMS error = 25.07. Result of filtering the noisy image in (b) using: (c) 3×3 mean, RMS error = 13.62; (d) 3×3 median, RMS error = 13.44; (e) 5×5 mean, RMS error = 16.17; (f) 5×5 median, RMS error = 13.47.

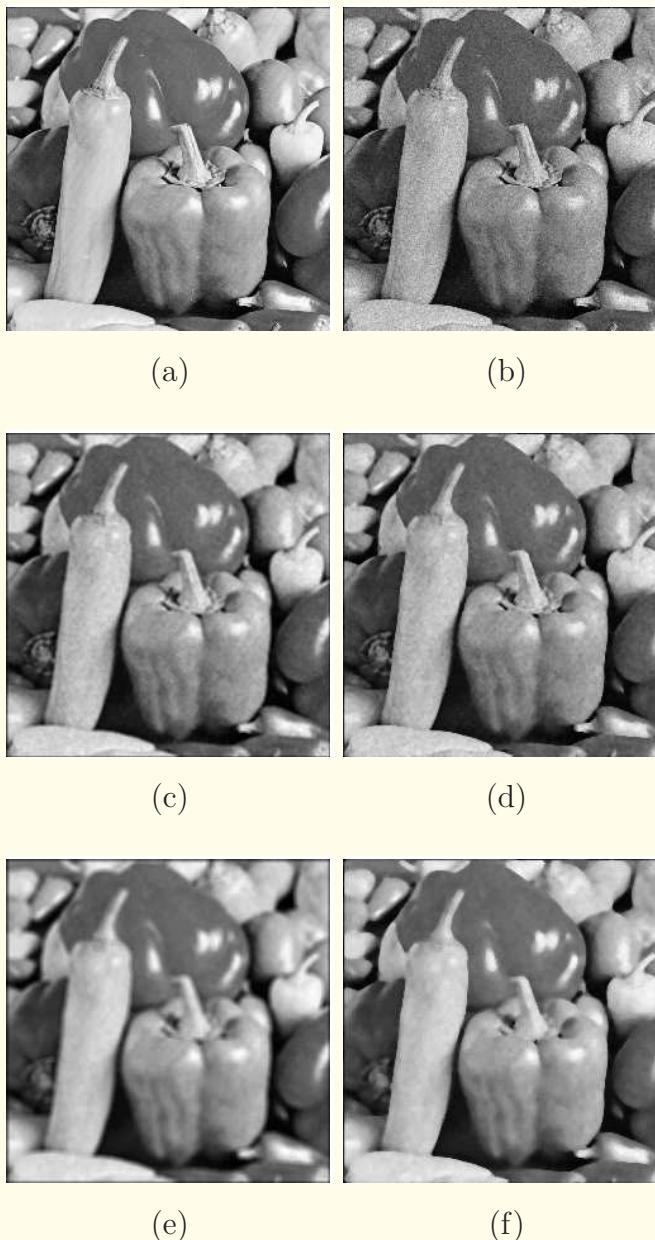


Figure 3.24: (a) Peppers test image. (b) Image in (a) with Poisson noise, RMS error = 10.94. Result of filtering the noisy image in (b) using: (c) 3×3 mean, RMS error = 11.22; (d) 3×3 median, RMS error = 8.56; (e) 5×5 mean, RMS error = 15.36; (f) 5×5 median, RMS error = 10.83.

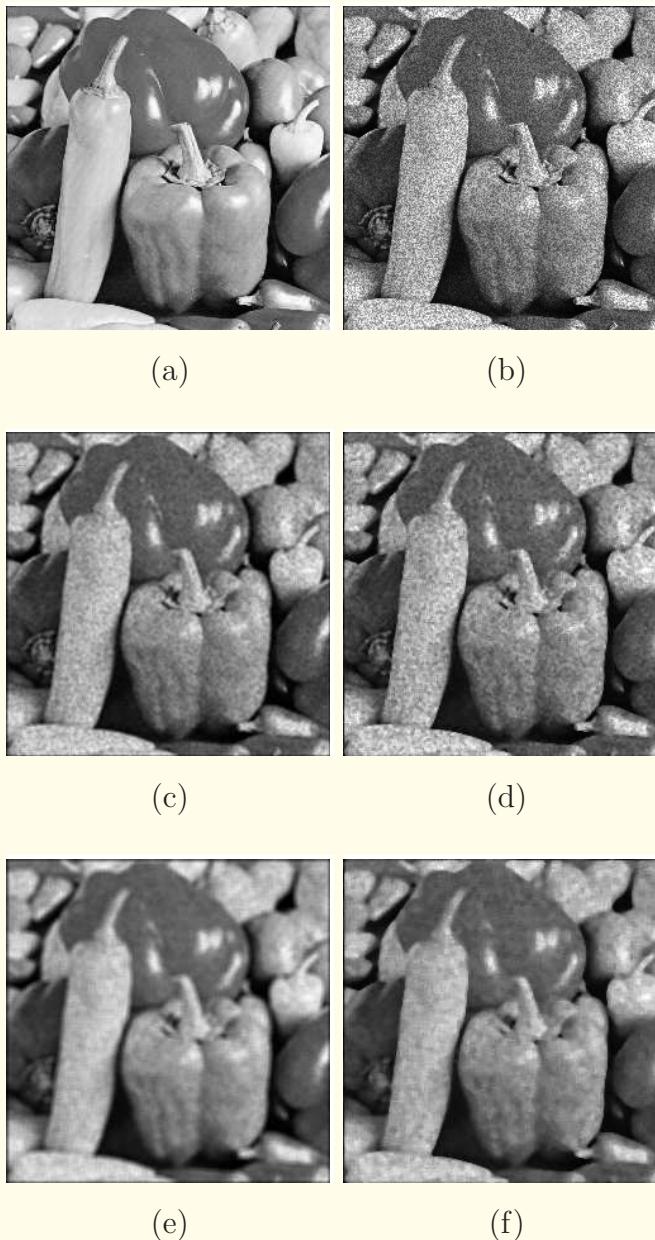


Figure 3.25: (a) Peppers test image. (b) Image in (a) with speckle noise, with $\mu = 0$, $\sigma^2 = 0.04$ (normalized), RMS error = 26.08. Result of filtering the noisy image in (b) using: (c) 3×3 mean, RMS error = 13.68; (d) 3×3 median, RMS error = 15.73; (e) 5×5 mean, RMS error = 16.01; (f) 5×5 median, RMS error = 14.66.

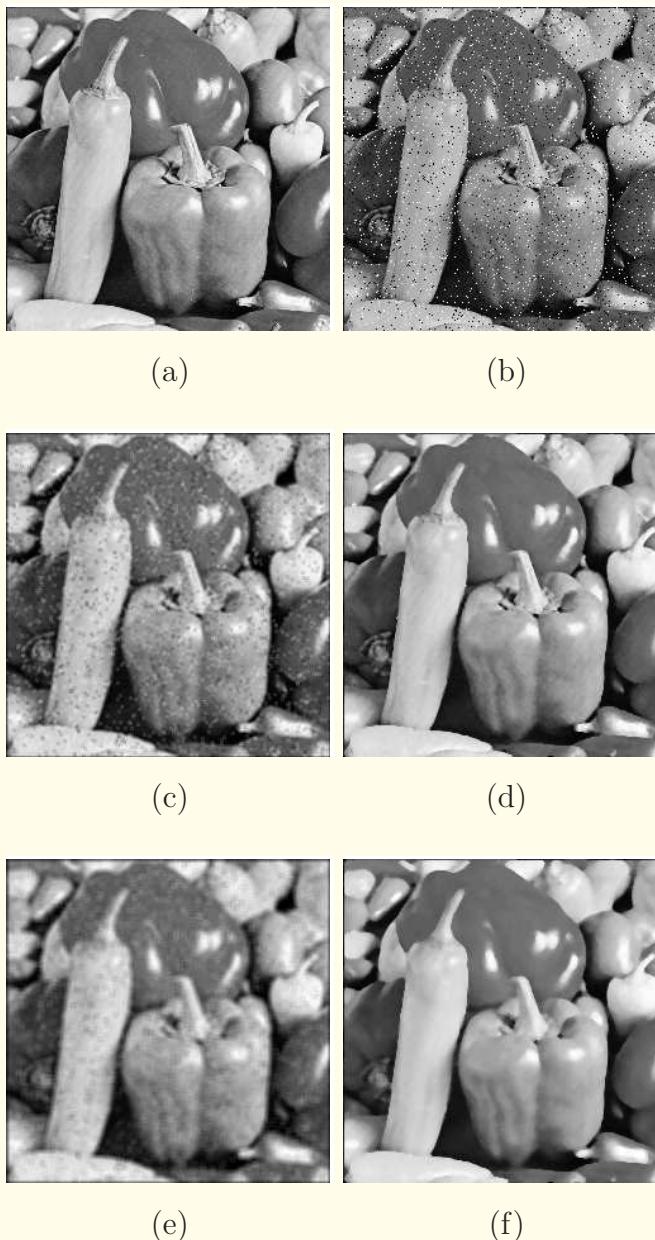


Figure 3.26: (a) Peppers test image. (b) Image in (a) with salt-and-pepper noise added, with density = 0.05, RMS error = 30.64. Result of filtering the noisy image in (b) using: (c) 3×3 mean, RMS error = 15.17; (d) 3×3 median, RMS error = 7.38; (e) 5×5 mean, RMS error = 16.96; (f) 5×5 median, RMS error = 10.41.

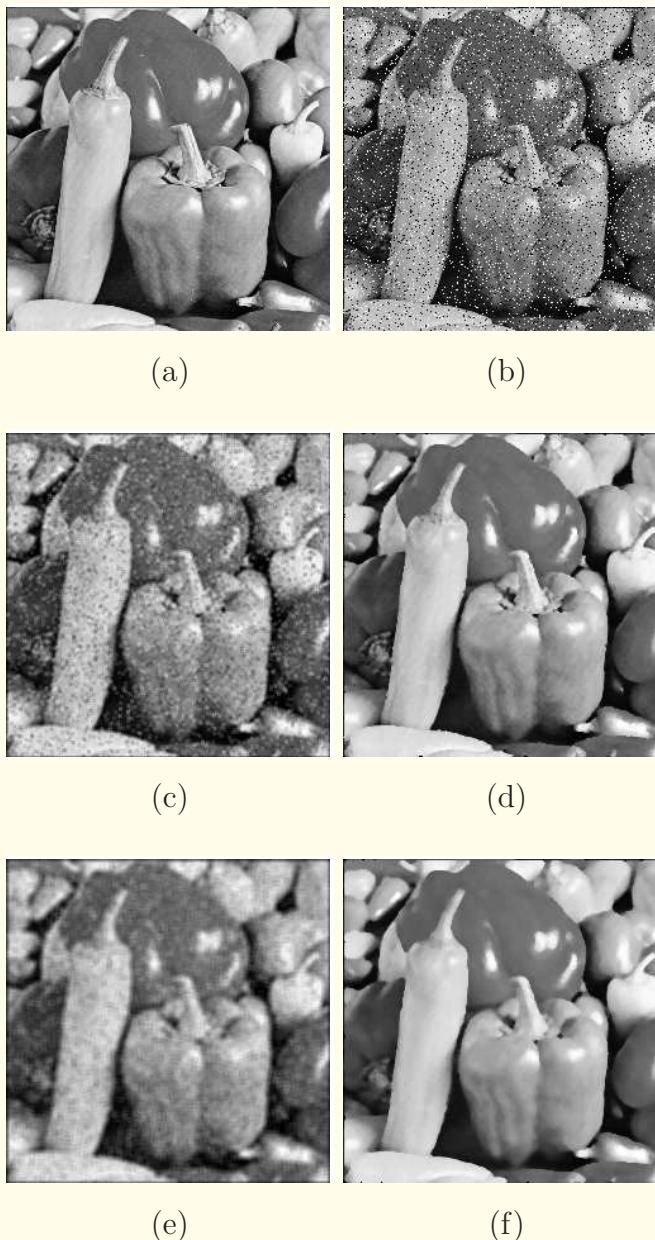


Figure 3.27: (a) Peppers test image. (b) Image in (a) with salt-and-pepper noise added, with density = 0.1, RMS error = 43.74. Result of filtering the noisy image in (b) using: (c) 3×3 mean, RMS error = 18.98; (d) 3×3 median, RMS error = 8.62; (e) 5×5 mean, RMS error = 18.71; (f) 5×5 median, RMS error = 11.11.



3.3.3 Order-statistic filters

Rank-order, from the minimum to the maximum, the pixel values in an appropriate neighborhood of the pixel being processed.

The i^{th} entry in the list is output of i^{th} order-statistic filter.

- ***Min filter:*** the first entry in the rank-ordered list, useful in removing high-valued impulse noise (isolated bright spots or “salt” noise).
- ***Max filter:*** the last entry in the rank-ordered list, useful in removing low-valued impulse noise (isolated dark spots or “pepper” noise).



- *Min/Max filter*: sequential application of the Min and Max filters, useful in removing salt-and-pepper noise.
- *Median filter*: the entry in the middle of the list.



- **α -trimmed mean filter:** the mean of a reduced list where the first α and the last α of the list is rejected, with $0 \leq \alpha < 0.5$. Outliers are rejected by the trimming process.
 $\alpha \approx 0.5$ rejects the entire list except the median or a few values close to it:
the output is close to or equal to that of the median filter.
The mean of the trimmed list provides a compromise between the generic mean and median filters.



- *L-filters*: a weighted combination of all of the elements in the rank-ordered list.

The use of appropriate weights can provide outputs equal to those of all of the filters listed above, and facilitate the design of several order-statistic-based nonlinear filters.



3.4 Frequency-domain Filters

Most images of natural entities scenes vary slowly and smoothly across space, and are usually devoid of step-like changes.

Such images have most of their energy concentrated in small regions around $(u, v) = (0, 0)$ in their spectra.

Uncorrelated random noise fields have a uniform, flat, or “white” spectrum, with an almost-constant energy level across the entire frequency space.

The SNR of a noisy, natural image is higher in low-frequency regions than in high-frequency regions.



Such images may be improved by suppressing high-frequency components beyond a certain cut-off frequency.

However, in removing high-frequency components, along with the noise, some desired image components will also be sacrificed.

Noise components in the low-frequency passband will continue to remain in the image.



Procedure for Fourier-domain filtering of an image $f(m, n)$:

1. Compute the 2D Fourier transform $F(k, l)$ of the image.
2. Design an appropriate 2D filter transfer function $H(k, l)$.
3. Obtain the filtered image (in the Fourier domain) as

$$G(k, l) = H(k, l) F(k, l). \quad (3.40)$$

Usually $H(k, l)$ is real, affecting only the magnitude of the input spectrum; phase unchanged.

With real-valued images, $H(k, l)$ should maintain (conjugate) symmetry; common to use isotropic filters.

$F(k, l)$ may have to be centered or folded.

4. Compute the inverse Fourier transform of $G(k, l)$.
If $F(k, l)$ was folded prior to filtering, it must be unfolded prior to the inverse transformation.
5. If the input image was zero-padded, trim the result $g(m, n)$.



3.4.1 Removal of high-frequency noise

Lowpass filters useful in removing high-frequency noise.

Assumption: noise is additive; Fourier components of the original image past a certain frequency cutoff are negligible.



Ideal lowpass filter:

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{otherwise.} \end{cases} \quad (3.41)$$

$D(u, v) = \sqrt{u^2 + v^2}$ is the distance of component at (u, v) from DC or $(u, v) = (0, 0)$, with the spectrum centered.

D_0 is the cutoff frequency, beyond which all components of the Fourier transform of the given image are set to zero.

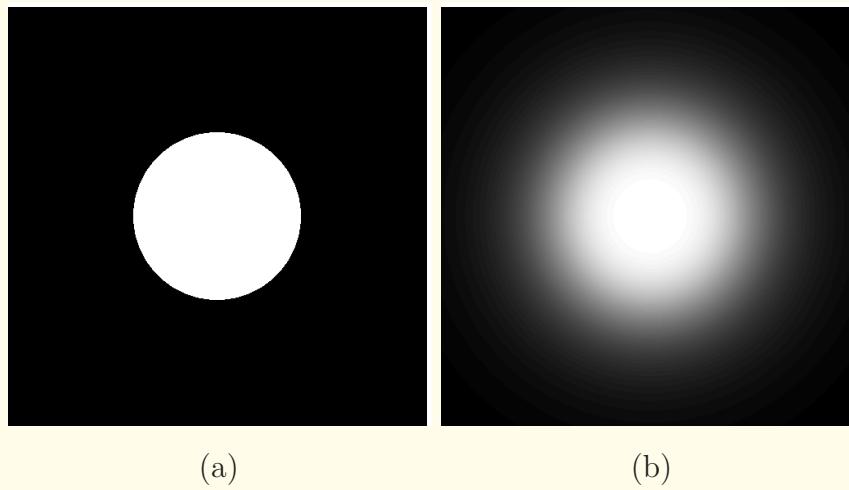


Figure 3.28: (a) The magnitude transfer function of an ideal lowpass filter. The cutoff frequency D_0 is 0.4 times the maximum frequency (that is, 0.2 times the sampling frequency). (b) The magnitude transfer function of a Butterworth lowpass filter, with normalized cutoff $D_0 = 0.4$ and order $n = 2$. The $(u, v) = (0, 0)$ point is at the center. The gain is proportional to the brightness (white represents 1.0 and black represents 0.0.)

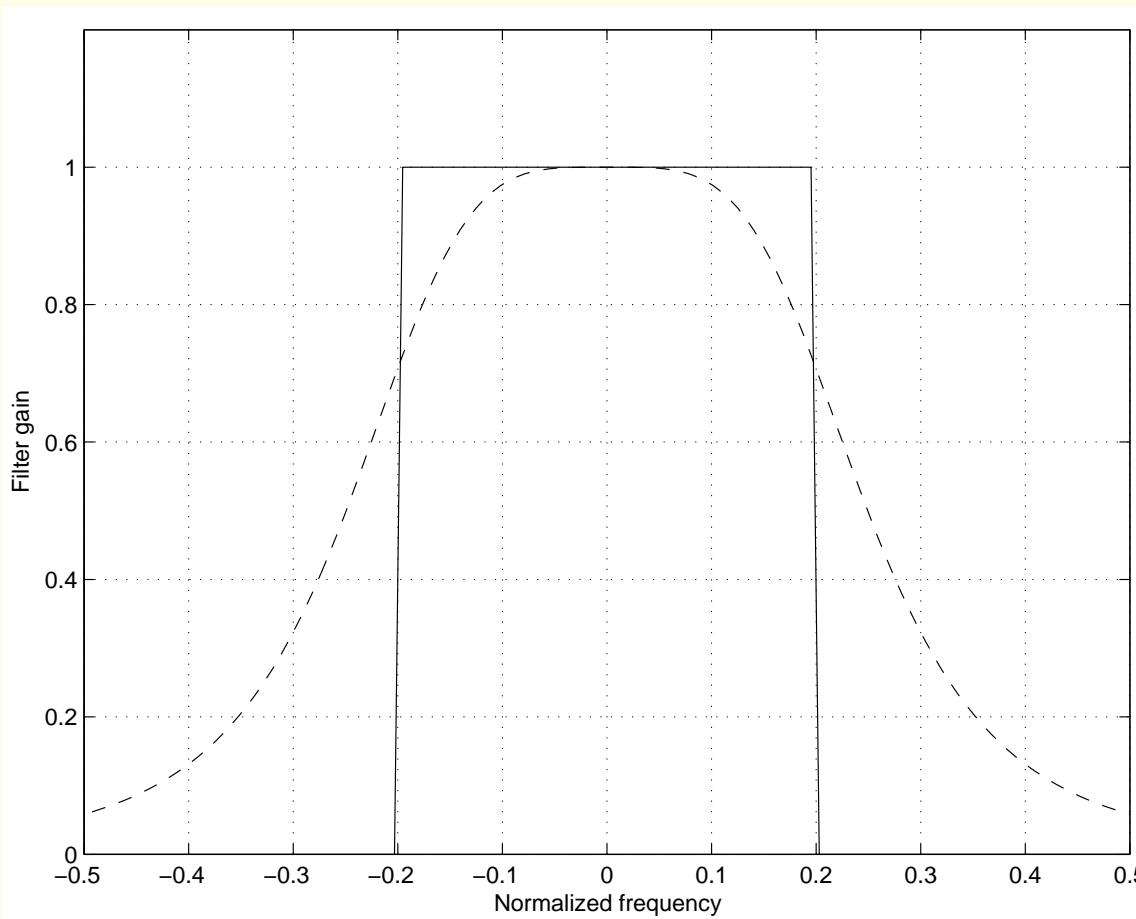


Figure 3.29: Profiles of the magnitude transfer functions of an ideal lowpass filter (solid line) and a Butterworth lowpass filter (dashed line), with normalized cutoff $D_0 = 0.4$ and order $n = 2$.



Glaring artifact in the result of the ideal filter:

faint echoes of the edges.

Inverse Fourier transform of circular ideal filter = Bessel function.

Multiplication of the Fourier transform of the image with the circle function is equivalent to convolution of the image in the space domain with the corresponding Bessel function.

The ripples or lobes of the Bessel function lead to echoes of strong edges, an artifact known as the *ringing artifact*.

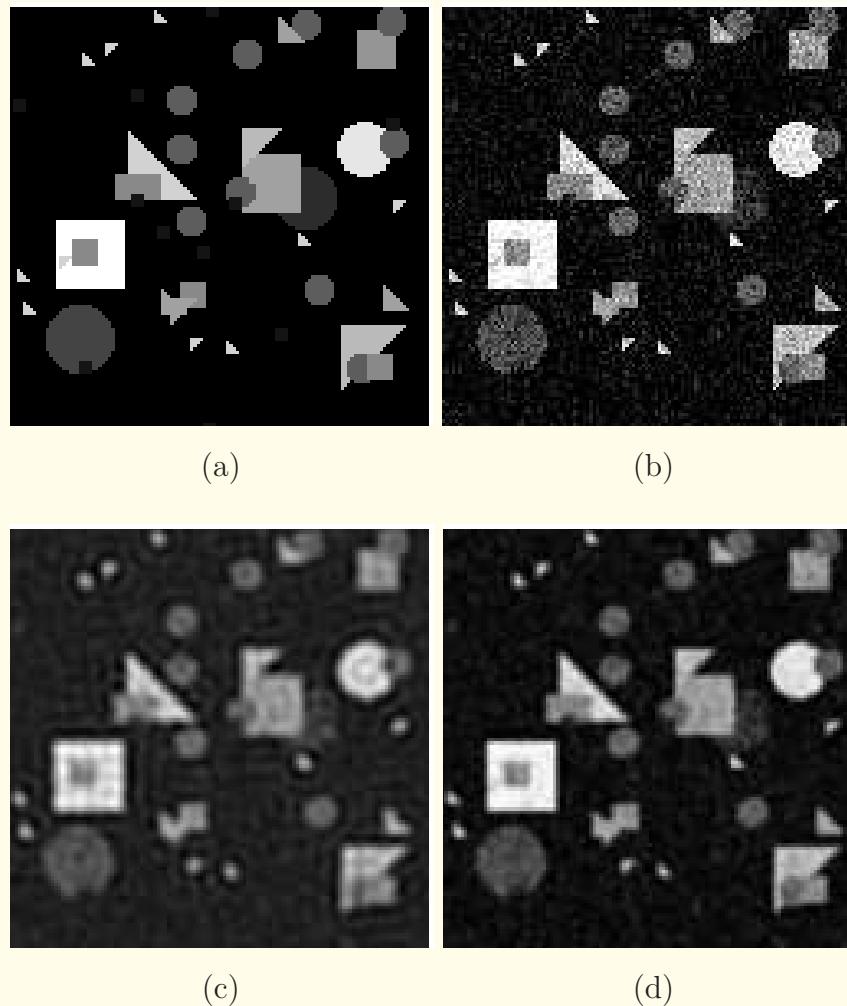
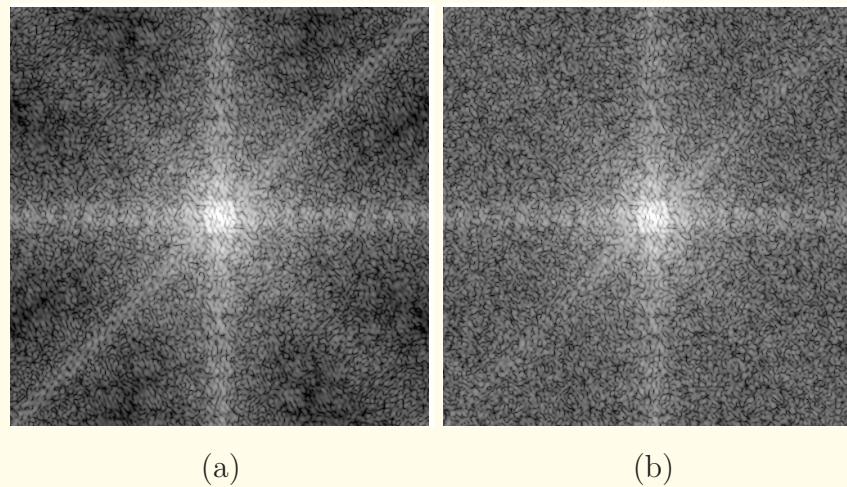


Figure 3.30: (a) The Shapes test image. (b) The test image with Gaussian noise having a normalized variance of 0.01 added. (c) The result of ideal lowpass filtering the noisy image, with normalized cutoff $D_0 = 0.4$; see Figure 3.28. (d) The result of filtering with a Butterworth lowpass filter having $D_0 = 0.4$ and order $n = 2$. See also Figure 3.31.



(a)

(b)

Figure 3.31: The centered (folded) Fourier log-magnitude spectrum of (a) the Shapes images in Figure 3.30 (a) and (b) the noisy Shapes image in Figure 3.30 (b).



The Butterworth lowpass filter:

Maximally flat magnitude response in the passband.

For a 1D Butterworth lowpass filter of order n , the first $2n - 1$ derivatives of the squared magnitude response are zero at $\omega = 0$.

Filter response is monotonic in the passband and the stopband.

2D Butterworth lowpass filter:

$$H(u, v) = \frac{1}{1 + (\sqrt{2} - 1) \left[\frac{D(u, v)}{D_0} \right]^{2n}}, \quad (3.42)$$



where n is the order of the filter,

$$D(u, v) = \sqrt{u^2 + v^2},$$

D_0 is the half-power radial cutoff frequency.

Gain = $\frac{1}{\sqrt{2}}$ at $D(u, v) = D_0$.

Transition from the passband to the stopband becomes steeper (faster) as the order n is increased.

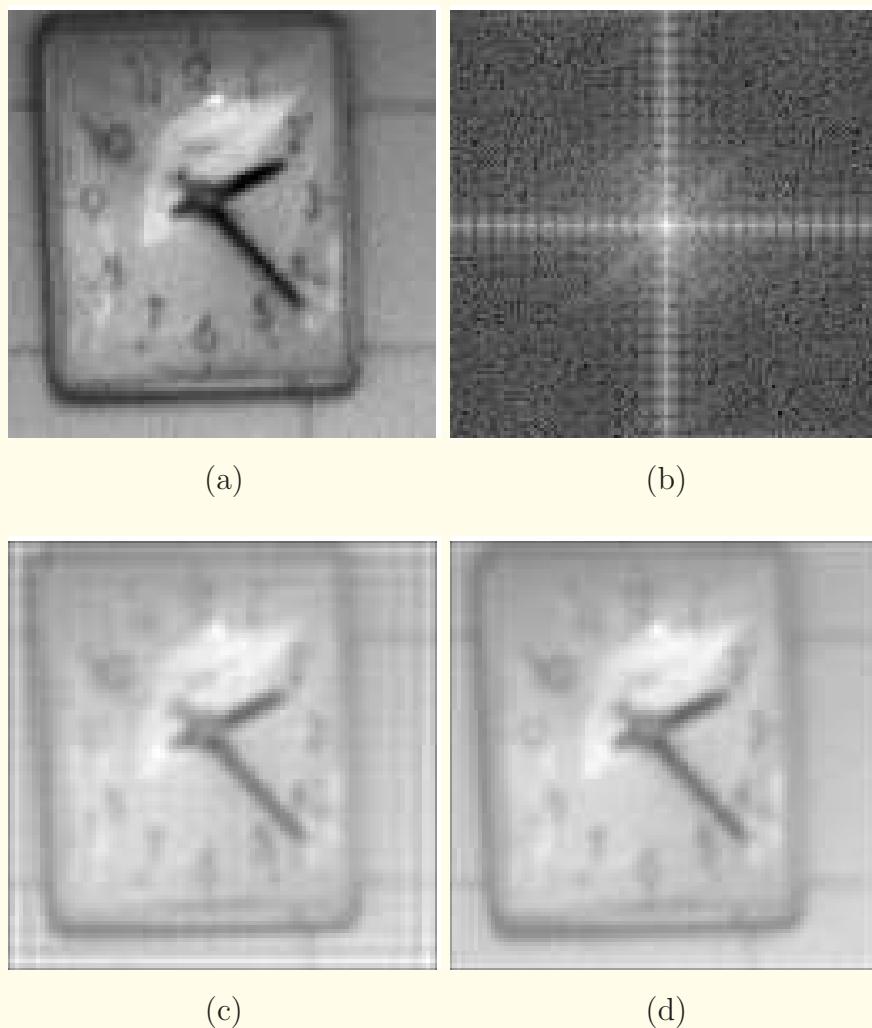


Figure 3.32: (a) Clock test image (101×101 pixels). (b) Log-magnitude spectrum of the image. (c) Result of the ideal lowpass filter, $D_0 = 0.4$. (d) Result of the Butterworth lowpass filter, with $D_0 = 0.4$ and order $n = 2$.



3.4.2 Removal of periodic artifacts

Periodic components in images give rise to impulse-like and periodic concentrations of energy in their Fourier spectra.

This characteristic facilitates the removal of periodic artifacts through selective band-reject, notch, or comb filtering.

Removing the artifactual components leads to the loss of the frequency-domain components of the desired image in the same regions, which could lead to some distortion in the filtered image.

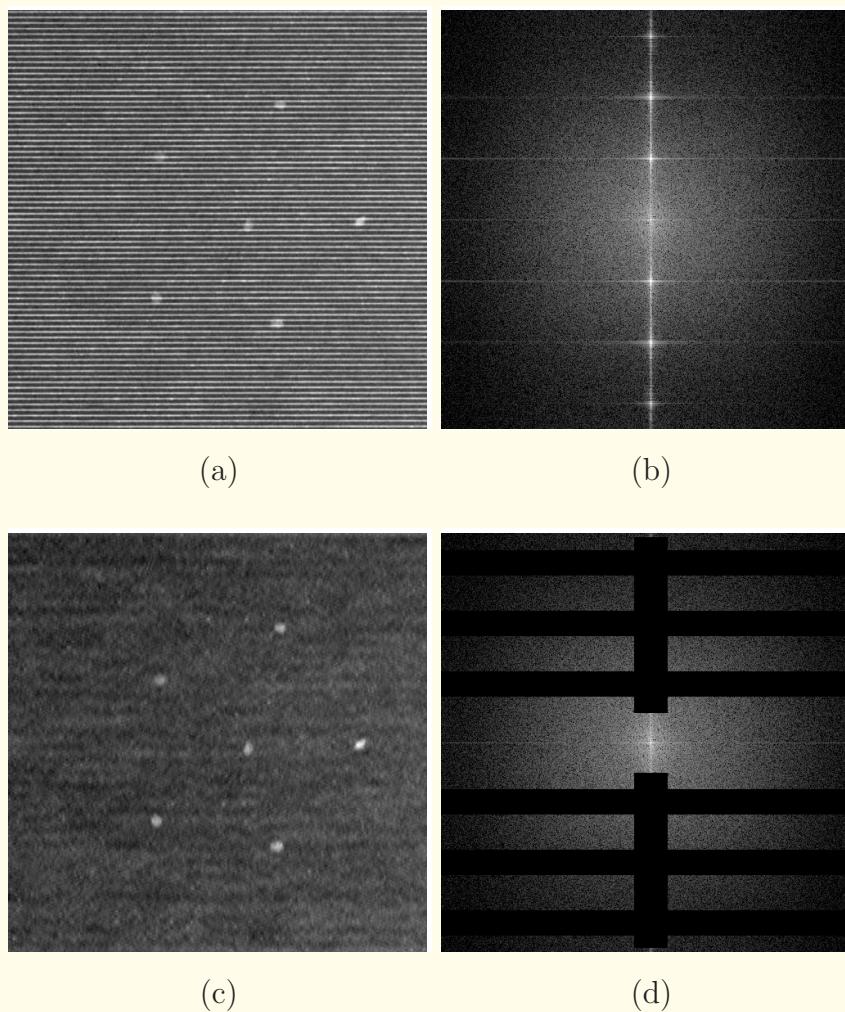
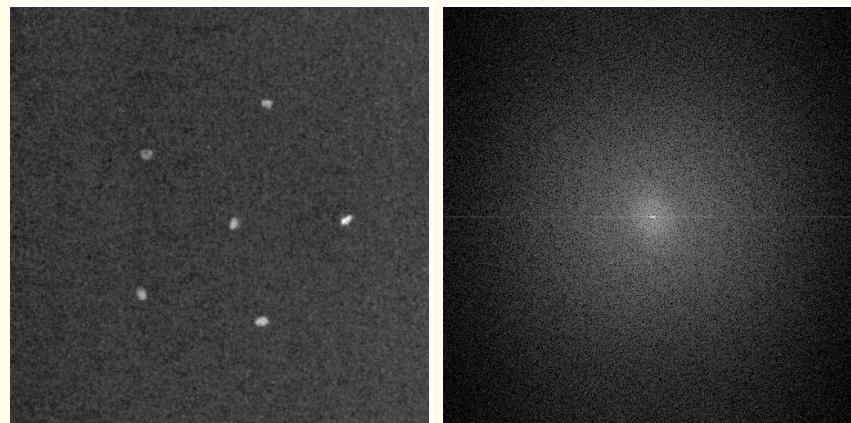


Figure 3.33: (a) Part of an image of a mammographic phantom with grid artifact; see also Figure 3.34. (b) Log-magnitude Fourier spectrum of the image in (a). (c) Filtered image. (d) Filtered version of the spectrum in (b). Phantom image courtesy of L.J. Hahn, Foothills Hospital, Calgary.



(a)

(b)

Figure 3.34: (a) Part of an image of a mammographic phantom with no grid artifact; compare with the image in Figure 3.33 (a). (b) Log-magnitude Fourier spectrum of the image in (a). Phantom image courtesy of L.J. Hahn, Foothills Hospital, Calgary.



3.5 Matrix Representation of Image Processing

Design of optimal filters and procedures requires the application of differentiation and statistical expectation to expressions involving images and filters.

The array or single-element form of representation of images and operations does not lend easily to such procedures.

Matrix representation of images, convolution, filters, and transforms facilitates efficient and compact expression of image processing, optimization, and estimation procedures.



3.5.1 Matrix representation of images

Sampled image represented by an $M \times N$ matrix:

$$\mathbf{f} = \{f(m, n) : m = 0, 1, 2, \dots, M - 1; \quad (3.43)$$

$$n = 0, 1, 2, \dots, N - 1\}.$$

M rows, each with N elements; the matrix has N columns.

Images are not merely arrays of numbers:

constraints imposed on matrix due to physical properties.



- Nonnegativity and upper bound: $f_{\min} \leq f(m, n) \leq f_{\max}$,
 f_{\min} and f_{\max} are the minimum and maximum values,
imposed by the characteristics of the object.
Usually, $f_{\min} = 0$, and $f(m, n) \geq 0$.
- Finite energy: $E_f = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^2(m, n) \leq E_{\max}$,
 E_{\max} is a finite limit on the total energy of the image.



- Smoothness: Images that represent real-life entities cannot change their characteristics abruptly.

$$\begin{aligned} f(m, n) - \\ \frac{1}{8} \left(\begin{array}{c} f(m-1, n-1) + f(m-1, n) + f(m-1, n+1) \\ + f(m, n-1) \qquad \qquad \qquad + f(m, n+1) \\ + f(m+1, n-1) + f(m+1, n) + f(m+1, n+1) \end{array} \right) \\ \leq S. \end{aligned} \tag{3.44}$$



$M \times N$ matrix converted to a vector by row ordering as

$$\mathbf{f} = [\mathbf{f}^1, \mathbf{f}^2, \dots, \mathbf{f}^M]^T \quad (3.45)$$

$$\mathbf{f}^m = [f(m, 1), f(m, 2), \dots, f(m, N)]^T$$

is the m^{th} row vector.

Column ordering may also be performed.



$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

(a)

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 1 \\ 4 \\ 7 \\ 2 \\ 5 \\ 8 \\ 3 \\ 6 \\ 9 \end{bmatrix}$$

(c)

Figure 3.35: (a) Matrix representation of a 3×3 image. Vector representation of the image in (a) by: (b) row ordering and (c) column ordering or stacking.



Using vector notation, energy of image

$$E = \mathbf{f}^T \mathbf{f} = \mathbf{f} \cdot \mathbf{f} = \sum_{i=1}^{MN} \mathbf{f}^2(i), \quad (3.46)$$

inner product or dot product of the vector with itself.

Energy also computed using the outer product as

$$E = Tr[\mathbf{f} \mathbf{f}^T], \quad (3.47)$$

$Tr[]$ is the trace (the sum of the main diagonal elements) of the resulting $MN \times MN$ matrix.



If the image elements are considered to be random variables, images may be treated as samples of stochastic processes.

- Mean: $\bar{\mathbf{f}} = E[\mathbf{f}]$, which is an $MN \times 1$ matrix or vector.



- Covariance: $\boldsymbol{\sigma} = E[(\mathbf{f} - \bar{\mathbf{f}})(\mathbf{f} - \bar{\mathbf{f}})^T] = E[\mathbf{f} \mathbf{f}^T] - \bar{\mathbf{f}} \bar{\mathbf{f}}^T$, which is an $MN \times MN$ matrix given by

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12} & \cdots & \sigma_{1P} \\ \sigma_{21} & \sigma_{22}^2 & \cdots & \sigma_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{P1} & \sigma_{P2} & \cdots & \sigma_{PP}^2 \end{bmatrix}, \quad (3.48)$$

where $P = MN$ and the matrix elements are

$$\sigma_{pq} = E[\{\mathbf{f}(p) - \bar{\mathbf{f}}(p)\}\{\mathbf{f}(q) - \bar{\mathbf{f}}(q)\}], \quad p, q = 1, 2, \dots, P. \quad (3.49)$$



σ_{pq} is the covariance between p^{th} and q^{th} elements of the image vector.

$\boldsymbol{\sigma}$ is symmetric: $\sigma_{pq} = \sigma_{qp}$.

Diagonal terms σ_{pp}^2 : variances of the elements of the image vector.



- Autocorrelation or scatter matrix $\phi = E[\mathbf{f} \ \mathbf{f}^T]$, which is an $MN \times MN$ matrix.
Note: $\phi = \sigma$ if $\bar{\mathbf{f}} = 0$.
- Normalized covariance: $\rho_{pq} = \frac{\sigma_{pq}}{(\sigma_{pp} \ \sigma_{qq})}$; $-1 \leq \rho_{pq} \leq 1$.

$$\boldsymbol{\rho} = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1P} \\ \rho_{21} & 1 & \cdots & \rho_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{P1} & \rho_{P2} & \cdots & 1 \end{bmatrix}. \quad (3.50)$$



The absolute scale of variation is retained in the diagonal standard deviation matrix:

$$\mathbf{D} = \begin{bmatrix} \sigma_{11} & 0 & \cdots & 0 \\ 0 & \sigma_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{PP} \end{bmatrix}. \quad (3.51)$$

$$\boldsymbol{\sigma} = \mathbf{D} \boldsymbol{\rho} \mathbf{D}.$$

Note: autocorrelation $\phi = \text{covariance } \boldsymbol{\sigma}$ if $\bar{\mathbf{f}} = 0$.



Two image vectors \mathbf{f} and \mathbf{g} are

- Uncorrelated if $E[\mathbf{f} \ \mathbf{g}^T] = E[\mathbf{f}] \ E[\mathbf{g}^T]$.

Then, the cross-covariance matrix $\boldsymbol{\sigma}_{fg}$ is a diagonal matrix, and the cross-correlation $\boldsymbol{\phi}_{fg} = \mathbf{I}$.

- Orthogonal if $E[\mathbf{f} \ \mathbf{g}^T] = \mathbf{0}$.
- Statistically independent if $p(\mathbf{f}, \mathbf{g}) = p(\mathbf{f}) \ p(\mathbf{g})$.

Then, \mathbf{f} and \mathbf{g} are uncorrelated.



3.5.2 Matrix representation of transforms

1D transforms: Signal $f(t)$ over the interval t_0 to $t_0 + T$ represented by an expansion of the form

$$f(t) = \sum_{k=0}^{\infty} a_k \varphi_k(t), \quad (3.52)$$

where the functions $\varphi_k(t)$ are mutually orthogonal, that is,

$$\int_{t_0}^{t_0+T} \varphi_k(t) \varphi_l^*(t) dt = \begin{cases} C & \text{if } k = l \\ 0 & \text{if } k \neq l. \end{cases} \quad (3.53)$$

The functions are said to be orthonormal if $C = 1$.



$$a_k = \frac{1}{C} \int_{t_0}^{t_0+T} f(t) \varphi_k^*(t) dt, \quad (3.54)$$

$k = 0, 1, 2, \dots, \infty;$

a_k is the projection of $f(t)$ on to $\varphi_k(t)$.



The set of functions $\{\varphi_k(t)\}$ is said to be complete or closed if there exists no square-integrable function $f(t)$ for which

$$\int_{t_0}^{t_0+T} f(t) \varphi_k^*(t) dt = 0, \quad \forall k; \quad (3.55)$$

i.e., $f(t)$ is orthogonal to all of the members of the set $\{\varphi_k(t)\}$.

If such a function exists, it should be a member of the set in order for the set to be closed or complete.

When $\{\varphi_k(t)\}$ is complete: *orthogonal basis*.



Fourier series representation of periodic signals uses an infinite set of sines and cosines at integral multiples (harmonics) of the fundamental frequency of the given signal.

$f(t)$ and $\varphi_k(t)$ must be square-integrable.



With a 1D signal as an $N \times 1$ vector or column matrix, we may represent transforms using an $N \times N$ matrix \mathbf{L} as

$$\mathbf{F} = \mathbf{L} \mathbf{f} \quad \text{and} \quad \mathbf{f} = \mathbf{L}^{*T} \mathbf{F}, \quad (3.56)$$

with $\mathbf{L} \mathbf{L}^{*T} = \mathbf{I}$.



The matrix operations are equivalent to

$$F(k) = \sum_{n=0}^{N-1} L(k, n) f(n) \text{ and } f(n) = \sum_{k=0}^{N-1} L^*(n, k) F(k), \quad (3.57)$$

for k or n going $0, 1, \dots, N - 1$, respectively.



DFT: define the matrix \mathbf{L} with its elements given by

$$L(k, n) = \exp\left(-j\frac{2\pi}{N} kn\right).$$

With the notation $W_N = \exp\left(-j\frac{2\pi}{N}\right)$,

we have $L(k, n) = W_N^{kn}$.

$$F(k) = \sum_{n=0}^{N-1} f(n) \exp\left(-j\frac{2\pi}{N} kn\right), \quad k = 0, 1, \dots, N-1. \quad (3.58)$$



$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ \vdots \\ F(N-2) \\ F(N-1) \end{bmatrix} =$$

$$\begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \dots & W_N^0 & W_N^0 \\ W_N^0 & W_N^{1 \times 1} & W_N^{1 \times 2} & \dots & W_N^{1 \times (N-2)} & W_N^{1 \times (N-1)} \\ W_N^0 & W_N^{2 \times 1} & W_N^{2 \times 2} & \dots & W_N^{2 \times (N-2)} & W_N^{2 \times (N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ W_N^0 & W_N^{(N-2) \times 1} & W_N^{(N-2) \times 2} & \dots & W_N^{(N-2) \times (N-2)} & W_N^{(N-2) \times (N-1)} \\ W_N^0 & W_N^{(N-1) \times 1} & W_N^{(N-1) \times 2} & \dots & W_N^{(N-1) \times (N-2)} & W_N^{(N-1) \times (N-1)} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(N-2) \\ f(N-1) \end{bmatrix}. \quad (3.59)$$



Because of the periodicity of the exponential function:

$$W_N^{m(N-1)} = W_N^{-m} = W_N^{N-m} \text{ for any integer } m;$$

$$W_N^N = 1.$$

For a given N , there are only N distinct functions:

$$W_N^k, k = 0, 1, 2, \dots, N - 1.$$

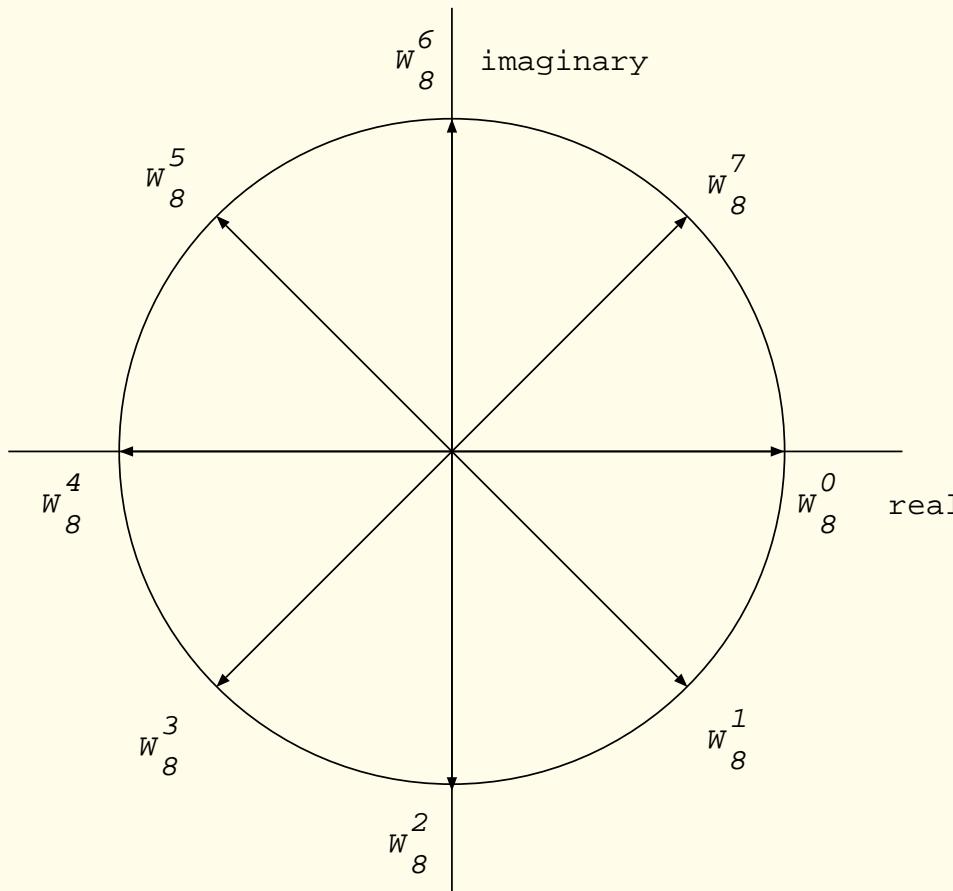


Figure 3.36: Vectors (or phasors) representing the $N = 8$ roots of unity, or W_8^k , $k = 0, 1, 2, \dots, 7$, where $W_8 = \exp\left(-j\frac{2\pi}{8}\right)$.



This property of the exponential function reduces the \mathbf{W} matrix to one with only N distinct values:

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ \vdots \\ F(N-2) \\ F(N-1) \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \cdots & W_N^{(N-2)} & W_N^{(N-1)} \\ W_N^0 & W_N^2 & W_N^4 & \cdots & W_N^{(N-4)} & W_N^{(N-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ W_N^0 & W_N^{(N-2)} & W_N^{(N-4)} & \cdots & W_N^4 & W_N^2 \\ W_N^0 & W_N^{(N-1)} & W_N^{(N-2)} & \cdots & W_N^2 & W_N^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(N-2) \\ f(N-1) \end{bmatrix}. \quad (3.60)$$



For $N = 8$, we get

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ F(4) \\ F(5) \\ F(6) \\ F(7) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ W^0 & W^2 & W^4 & W^6 & W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^1 & W^4 & W^7 & W^2 & W^5 \\ W^0 & W^4 & W^0 & W^4 & W^0 & W^4 & W^0 & W^4 \\ W^0 & W^5 & W^2 & W^7 & W^4 & W^1 & W^6 & W^3 \\ W^0 & W^6 & W^4 & W^2 & W^0 & W^6 & W^4 & W^2 \\ W^0 & W^7 & W^6 & W^5 & W^4 & W^3 & W^2 & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}, \quad (3.61)$$

where the subscript $N = 8$ to W has been suppressed.



2D transforms:

$$F(k, l) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m, n) \varphi(m, n, k, l), \quad (3.62)$$

$$f(m, n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k, l) \psi(m, n, k, l), \quad (3.63)$$

where $\varphi(m, n, k, l)$ is the forward transform kernel,

$\psi(m, n, k, l)$ is the inverse transform kernel.



The kernel is said to be separable if

$$\varphi(m, n, k, l) = \varphi_1(m, k) \varphi_2(n, l),$$

and symmetric if φ_1 and φ_2 are functionally equal.



Then, the 2D transform may be computed in two simpler steps:

1D row transforms followed by 1D column transforms.

$$F_1(m, l) = \sum_{n=0}^{N-1} f(m, n) \varphi(n, l); \quad m, l = 0, 1, \dots, N - 1; \quad (3.64)$$

$$F(k, l) = \sum_{m=0}^{N-1} F_1(m, l) \varphi(m, k); \quad k, l = 0, 1, \dots, N - 1. \quad (3.65)$$



In the case of the 2D Fourier transform, we have the kernel

$$\begin{aligned}\varphi(m, n, k, l) &= \exp\left[-j\frac{2\pi}{N}(mk + nl)\right] \quad (3.66) \\ &= \exp\left[-j\frac{2\pi}{N}mk\right] \exp\left[-j\frac{2\pi}{N}nl\right].\end{aligned}$$

The kernel is separable and symmetric.



The 2D DFT may be expressed as

$$\mathbf{F} = \mathbf{W} \mathbf{f} \mathbf{W}, \quad (3.67)$$

where \mathbf{f} is the $N \times N$ image matrix,

and \mathbf{W} is a symmetric $N \times N$ matrix with

$$W_N^{km} = \exp \left[-j \frac{2\pi}{N} km \right];$$

Periodicity of W_N : matrix \mathbf{W} has only N distinct values.



The DFT matrix \mathbf{W} is symmetric, with its rows and columns being mutually orthogonal:

$$\sum_{m=0}^{N-1} W_N^{mk} W_N^{ml*} = \begin{cases} N & k = l \\ 0 & k \neq l \end{cases}. \quad (3.68)$$

Then, $\mathbf{W}^{-1} = \frac{1}{N}\mathbf{W}^*$, which leads to

$$\mathbf{f} = \frac{1}{N^2} \mathbf{W}^* \mathbf{F} \mathbf{W}^*. \quad (3.69)$$



A number of transforms such as the Fourier, Walsh–Hadamard, and discrete cosine may be expressed as

$$\mathbf{F} = \mathbf{A} f \mathbf{A},$$

with the matrix \mathbf{A} constructed using the relevant basis functions.

The transform matrices may be decomposed into products of matrices with fewer nonzero elements:

reducing redundancy and computational requirements.

The DFT matrix may be factored into a product of $2 \ln N$ sparse and diagonal matrices, leading to the FFT algorithm.



The Walsh–Hadamard Transform:

The orthonormal, complete set of 1D Walsh functions defined over the interval $0 \leq x \leq 1$ is given by the iterative relationships

$$\varphi_n(x) = \begin{cases} \varphi_{\left[\frac{n}{2}\right]}(2x), & x < \frac{1}{2}, \\ \varphi_{\left[\frac{n}{2}\right]}(2x - 1), & x \geq \frac{1}{2}, \text{ } n \text{ odd,} \\ -\varphi_{\left[\frac{n}{2}\right]}(2x - 1), & x \geq \frac{1}{2}, \text{ } n \text{ even,} \end{cases} \quad (3.70)$$

where $\left[\frac{n}{2}\right]$ is the integral part of $\frac{n}{2}$,



$$\varphi_0(x) = 1, \quad (3.71)$$

$$\varphi_1(x) = \begin{cases} 1 & x < \frac{1}{2} \\ -1 & x \geq \frac{1}{2}. \end{cases} \quad (3.72)$$

The n^{th} function φ_n is generated by compression of the function $\varphi_{\left[\frac{n}{2}\right]}$ into its first half and $\pm\varphi_{\left[\frac{n}{2}\right]}$ into its second half.

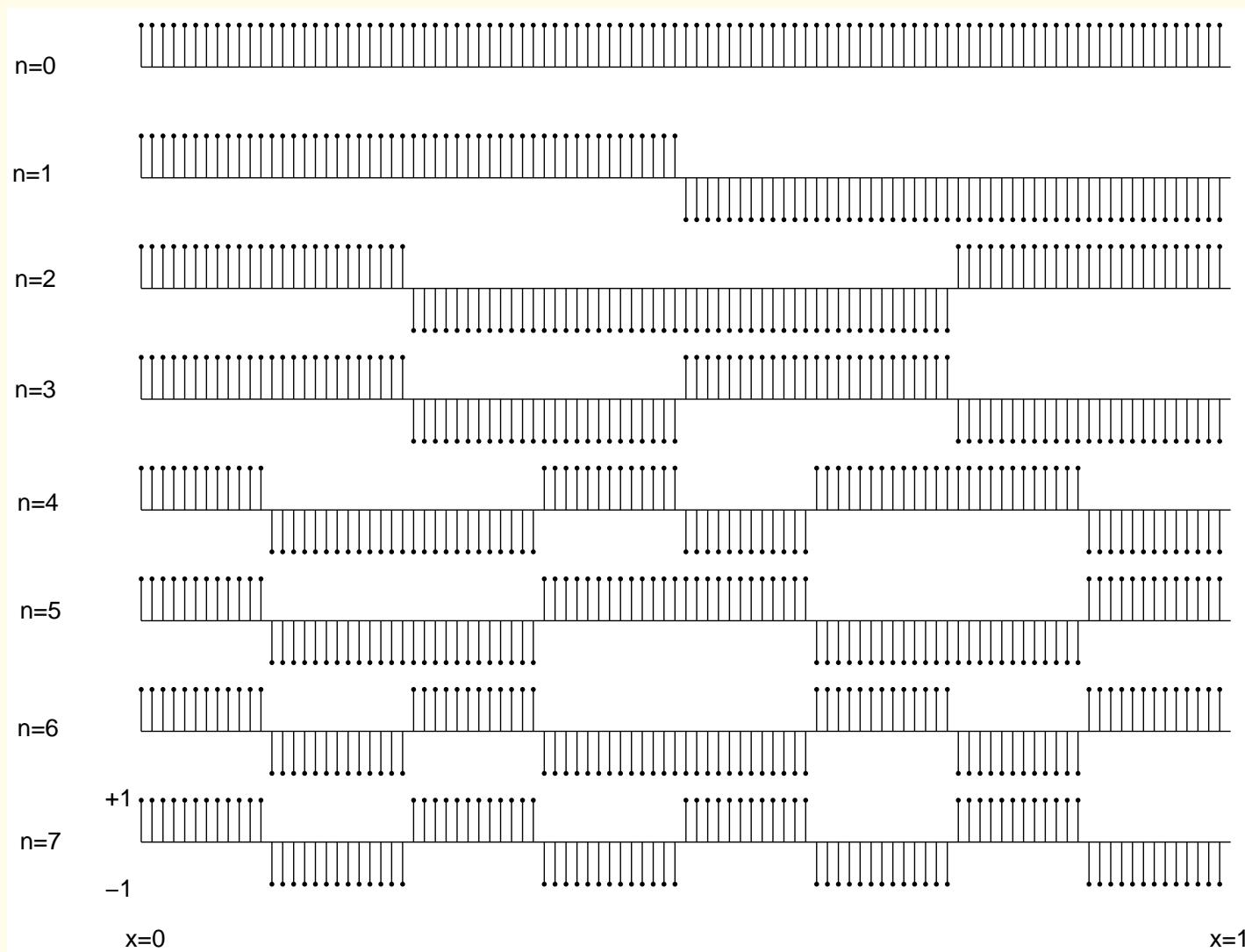


Figure 3.37: The first eight Walsh functions sampled with 100 samples over the interval $(0, 1)$.



Walsh functions are ordered by the number of zero-crossings in the interval $(0, 1)$, called *sequency*.

If the Walsh functions with the number of zero-crossings $\leq (2^n - 1)$ are sampled with $N = 2^n$ uniformly spaced points, we get a square matrix representation that is orthogonal.

Rows ordered with increasing number of zero-crossings.

The ordering of the Walsh functions varies with the formulation.



$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \quad (3.73)$$

Walsh transform of a 1D signal \mathbf{f} :

$$\mathbf{F} = \mathbf{A} \mathbf{f}.$$



Another formulation of the Walsh transform:

$$\varphi(n, k) = \frac{1}{N} \prod_{p=0}^{P-1} (-1)^{[b_p(n)b_{P-1-p}(k)]}, \quad (3.74)$$

$$\varphi(m, n, k, l) = \frac{1}{N} \prod_{p=0}^{P-1} (-1)^{[b_p(m)b_{P-1-p}(k) + b_p(n)b_{P-1-p}(l)]}, \quad (3.75)$$

$b_p(m)$ is the p^{th} bit in the P -bit binary representation of m .



Major advantage of the Walsh transform:

Kernel has integral values of $+1$ and -1 only.

Only addition and subtraction of the input image pixels.

Identical operations in forward and inverse transformation.



Except for the ordering of rows, the discrete Walsh matrices are equivalent to the Hadamard matrices of rank 2^n :

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad (3.76)$$

$$\mathbf{A}_{2N} = \begin{bmatrix} \mathbf{A}_N & \mathbf{A}_N \\ \mathbf{A}_N & -\mathbf{A}_N \end{bmatrix}. \quad (3.77)$$



Then, by defining $\mathbf{A} = \frac{1}{\sqrt{N}} \mathbf{A}_N$,

the Walsh–Hadamard transform (WHT) of a 2D function may be expressed as

$$\mathbf{F} = \mathbf{A} \mathbf{f} \mathbf{A}, \text{ and } \mathbf{f} = \mathbf{A} \mathbf{F} \mathbf{A}, \quad (3.78)$$

where all matrices are of size $N \times N$.

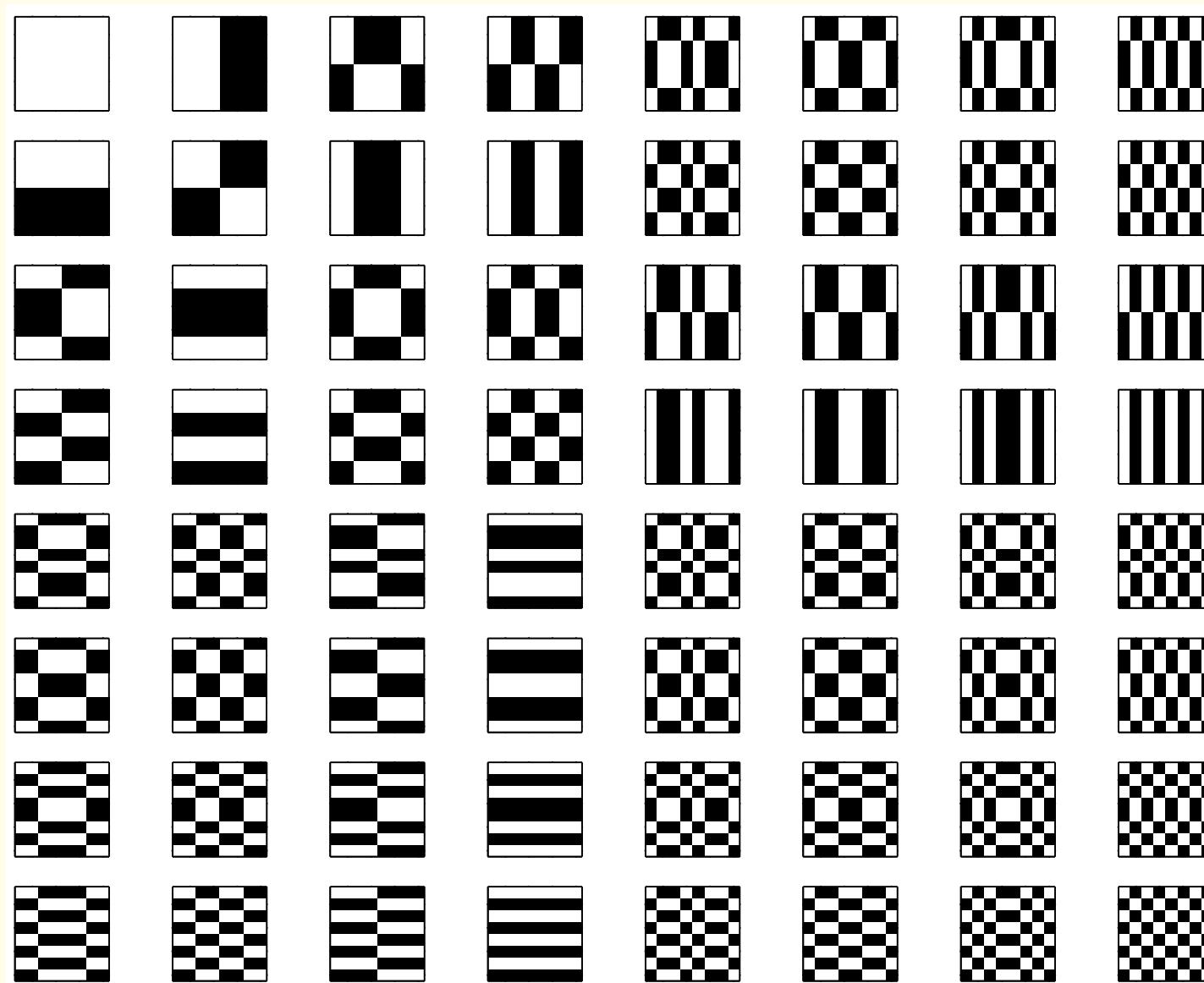


Figure 3.38: The first 64 Walsh–Hadamard 2D basis functions. Black represents a pixel value of -1 , and white $+1$. Each function was computed as an 8×8 matrix.



3.5.3 *Matrix representation of convolution*

With images represented as vectors, linear system operations such as convolution and filtering may be represented as *matrix–vector multiplications*.

Let us first consider the 1D LSI system.

Assume the system to be causal, and to have an infinite impulse response (IIR).

The input-output relationship is given by the linear convolution

$$g(n) = \sum_{\alpha=0}^n f(\alpha) h(n - \alpha). \quad (3.79)$$



If the input is given over N samples, we could represent the output over the interval $[0, N]$ as

$$\mathbf{g} = \mathbf{h} \mathbf{f}, \quad (3.80)$$

$$\begin{bmatrix} g(0) \\ g(1) \\ g(2) \\ \vdots \\ g(N) \end{bmatrix} = \begin{bmatrix} h(0) & 0 & \cdots & \cdots & 0 \\ h(1) & h(0) & 0 & \cdots & 0 \\ h(2) & h(1) & h(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h(N) & h(N-1) & h(N-2) & \cdots & h(0) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(N) \end{bmatrix}. \quad (3.81)$$

\mathbf{h} is a Toeplitz-like matrix (computational advantages).

Toeplitz: square matrix, elements are equal along every diagonal.

There will be zeros in the lower-left portion of \mathbf{h} if $h(n)$ has fewer samples than $f(n)$ and $g(n)$: \mathbf{h} is banded.



If the impulse response is of a finite duration of $M + 1$ samples (finite impulse response — FIR filter):

$$g(n) = \sum_{\alpha=n-\frac{M}{2}}^{n+\frac{M}{2}} f(\alpha) h(n - \alpha), \quad (3.82)$$

$$g(n) = \sum_{\alpha=0}^M f\left(\alpha + n - \frac{M}{2}\right) h\left(\frac{M}{2} - \alpha\right). \quad (3.83)$$

This relationship may also be expressed as $\mathbf{g} = \mathbf{h} \mathbf{f}$, with the matrix and vectors constructed as in Figure 3.39.



\mathbf{h} is now banded and Toeplitz-like.

Each row of \mathbf{h} , except the first, is a right-shifted version of the preceding row.

It has been assumed that $M < N$.



$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(N) \end{bmatrix} =$$
$$\begin{bmatrix} h\left(\frac{M}{2}\right) & h\left(\frac{M}{2} - 1\right) & \cdots & h(0) & \cdots & h\left(1 - \frac{M}{2}\right) & h\left(-\frac{M}{2}\right) & 0 & \cdots & 0 \\ 0 & h\left(\frac{M}{2}\right) & h\left(\frac{M}{2} - 1\right) & \cdots & h(0) & \cdots & h\left(1 - \frac{M}{2}\right) & h\left(-\frac{M}{2}\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \cdots & \ddots & \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & h\left(\frac{M}{2}\right) & h\left(\frac{M}{2} - 1\right) & \cdots & h(0) & \cdots & \cdots & h\left(1 - \frac{M}{2}\right) & h\left(-\frac{M}{2}\right) \end{bmatrix} \begin{bmatrix} f\left(-\frac{M}{2}\right) \\ f\left(1 - \frac{M}{2}\right) \\ \vdots \\ f\left(N + \frac{M}{2}\right) \end{bmatrix}$$

Figure 3.39: Construction of the matrix and vectors for convolution in the case of an FIR filter.



Periodic or circular convolution:

All of the signals are assumed to be of finite duration and periodic, with the period being equal to N samples.

Periodic shifting: samples that go out of the frame of N samples at one end will reappear at the other end.

$$g_p(n) = \sum_{\alpha=0}^{N-1} f_p(\alpha) h_p([n - \alpha] \bmod N), \quad (3.84)$$

subscript p : periodic version or interpretation of signals.



The result of periodic convolution of two periodic signals of N samples each is a periodic signal of the same period N ;

result of linear convolution would have $2N - 1$ samples.

Periodic convolution may be used to achieve the same result as linear convolution by padding the signals with zeros so as to increase their duration to at least $2N - 1$ samples.

Implementing convolution as the inverse DFT of the product of the DFTs of the two signals: periodic convolution.

Zero padding will be necessary if linear convolution is desired.

Causal LSI system: *linear* convolution.



Periodic convolution may also be expressed as

$$\mathbf{g} = \mathbf{h} \mathbf{f},$$

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(N-2) \\ g(N-1) \end{bmatrix} = \begin{bmatrix} h(0) & h(N-1) & \cdots & h(2) & h(1) \\ h(1) & h(0) & \cdots & h(3) & h(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h(N-2) & h(N-3) & \cdots & h(0) & h(N-1) \\ h(N-1) & h(N-2) & \cdots & h(1) & h(0) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-2) \\ f(N-1) \end{bmatrix} \quad (3.85)$$

Each row of \mathbf{h} is a right-circular shift of the previous row, and the matrix is square: *circulant matrix*.



3.5.4 Illustrations of convolution

Convolution in 1D:

$$g(n) = \sum_{k=0}^n f(k) h(n - k). \quad (3.86)$$

h needs to be reversed with respect to the index of summation k , and shifted by n where the output sample is to be computed.

The index n may be run over a certain range of interest or over all time for which the result $g(n)$ exists.



For each n , the reversed and shifted version of h is multiplied with the signal f on a point-by-point basis and summed.

The multiplication and summation operation, together, are comparable to the dot product operation, performed over the nonzero overlapping parts of the two signals.



Example:

$f(n) = [4, 1, 3, 1]$, defined for $n = 0, 1, 2, 3$.

$h(n) = [3, 2, 1]$, for $n = 0, 1, 2$.

Result g has more samples (longer duration) than f or h :

the result of linear convolution of two signals with N_1 and N_2 samples will have $N_1 + N_2 - 1$ samples.



n: 0 1 2 3 4 5 6 7

$f(n)$: 4 1 3 1

$h(n)$: 3 2 1 0

k: 0 1 2 3 4 5 6 7

$f(k)$: 4 1 3 1 0 0 0 0

$h(0-k)$: 0 1 2 3

$h(1-k)$: 0 1 2 3

$h(2-k)$: 0 1 2 3

$h(3-k)$: 0 1 2 3

$h(4-k)$: 0 1 2 3

$h(5-k)$: 0 1 2 3

$h(6-k)$: 0 1 2 3

$g(n)$: 12 11 15 10 5 1 0 0



n: 0 1 2 3 4 5 6 7

Figure 3.40: Illustration of the linear convolution of two 1D signals. Observe the reversal of $h(n)$, shown as $h(0 - k)$, and the shifting of the reversed signal, shown as $h(1 - k)$, $h(2 - k)$, etc.



In the matrix notation of Equation 3.81, the convolution example above is expressed as

$$\begin{bmatrix} 3 & 0 & 0 & 0 & 1 & 2 \\ 2 & 3 & 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 0 \\ 0 & 0 & 1 & 2 & 3 & 0 \\ 0 & 0 & 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \\ 3 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 12 \\ 11 \\ 15 \\ 10 \\ 5 \\ 1 \end{bmatrix}. \quad (3.87)$$

The result is identical to that shown in Figure 3.40.



Convolution in 2D:

The output of an LSI imaging or image processing system is given as the convolution of the input image with the PSF:

$$g(m, n) = \sum_{\alpha=0}^{N-1} \sum_{\beta=0}^{N-1} f(\alpha, \beta) h(m - \alpha, n - \beta). \quad (3.88)$$

For the sake of generality, the range of summation is allowed to span the full spatial range of the output image.



When the filter PSF is of a much smaller spatial extent than the input image, it becomes convenient to locate the origin $(0, 0)$ at the center of the PSF,

and use positive and negative indices to represent the omnidirectional (and noncausal) nature of the PSF.

$$g(m, n) = \sum_{\alpha=-M}^{M} \sum_{\beta=-M}^{M} f(\alpha, \beta) h(m - \alpha, n - \beta), \quad (3.89)$$

size of PSF assumed to be odd: $(2M + 1) \times (2M + 1)$.

In this format, 2D convolution may be interpreted as a mask operation performed on the input image.



The PSF is reversed (flipped or reflected) about both of its axes,

placed on top of the input image at the coordinate where the output value is to be computed,

a point-by-point multiplication is performed of the overlapping areas of the two functions,

and the resulting products are added.

The operation needs to be performed at every spatial location for which the output exists, by dragging and placing the reversed PSF at every pixel of the input image.

In the case of a PSF having symmetry about both of its axes, the reversal step has no effect and is not required.



Reflect (reverse)
about the vertical

1	4	7
2	5	8
3	6	9

(a)

Reflect (reverse)
about the horizontal

7	4	1
8	5	2
9	6	3

(b)

9	6	3
8	5	2
7	4	1

(c)

1	2	1	1	1	2	2	1
3	1	6	1	3	2	3	2
1	2	5	2	1	2	1	3
3	2	4	1	3	2	1	0
1	3	2	4	1	1	1	0
4	2	3	1	9	2	6	3
2	3	1	4	2	4	3	1
1	2	1	1	0	1	0	1

(d)

1	6	16	19	12	13	17	23	18	7
5	22	46	36	33	45	55	60	57	22
10	35	72	67	62	72	90	94	95	39
14	44	90	79	93	96	93	90	86	41
10	38	88	105	107	98	63	62	51	26
15	53	114	117	119	106	69	75	39	16
13	50	108	113	103	110	87	90	49	15
17	52	106	95	82	89	88	96	50	24
8	30	59	68	52	67	44	46	20	17
3	12	24	27	15	12	6	12	6	9

(e)



Figure 3.41: Illustration of the linear convolution of two 2D functions. Observe the reversal of the PSF in parts (a) – (c), and the shifting of the reversed PSF as a mask placed on the image to be filtered, in part (d). The shifted mask is shown for two pixel locations. Observe that the result needs to be written in a different array. The result, of size 10×10 and shown in part (e), has two rows and two columns more than the input image (of size 8×8).



3.5.5 Diagonalization of a circulant matrix

A circulant matrix is diagonalized by the DFT.

Consider the general circulant matrix

$$\mathbf{C} = \begin{bmatrix} C(0) & C(1) & C(2) & \cdots & C(N-2) & C(N-1) \\ C(N-1) & C(0) & C(1) & \cdots & C(N-3) & C(N-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ C(2) & C(3) & C(4) & \cdots & C(0) & C(1) \\ C(1) & C(2) & C(3) & \cdots & C(N-1) & C(0) \end{bmatrix}. \quad (3.90)$$



Let $W = \exp(j\frac{2\pi}{N})$.

$W^{kN} = 1$ for any integer k .

$W^k, k = 0, 1, 2, \dots, N - 1$, are the N distinct roots of unity.



Now, consider

$$\lambda(k) = C(0) + C(1)W^k + C(2)W^{2k} + \cdots + C(N-1)W^{(N-1)k}. \quad (3.91)$$

It follows that

$$\lambda(k)W^k = C(N-1) + C(0)W^k + C(1)W^{2k} + \cdots + C(N-2)W^{(N-1)k},$$

$$\lambda(k)W^{2k} = C(N-2) + C(N-1)W^k + C(0)W^{2k} + \cdots + C(N-3)W^{(N-1)k},$$

⋮

$$\lambda(k)W^{(N-1)k} = C(1) + C(2)W^k + C(3)W^{2k} + \cdots + C(0)W^{(N-1)k}.$$



$$\lambda(k) \mathbf{W}(k) = \mathbf{C} \mathbf{W}(k), \quad (3.92)$$

where

$$\mathbf{W}(k) = [1, W^k, W^{2k}, \dots, W^{(N-1)k}]^T. \quad (3.93)$$

$\lambda(k)$ is an *eigenvalue* and

$\mathbf{W}(k)$ is an *eigenvector* of the circulant matrix \mathbf{C} .



Because there are N values W^k , $k = 0, 1, \dots, N - 1$, that are distinct, there are N distinct eigenvectors $\mathbf{W}(k)$, which may be written as the $N \times N$ matrix

$$\mathbf{W} = [\mathbf{W}(0) \ \mathbf{W}(1) \ \dots \ \mathbf{W}(N - 1)], \quad (3.94)$$

which is related to the DFT.

The $(n, k)^{th}$ element of \mathbf{W} is $\exp(j\frac{2\pi}{N}nk)$.



Due to the orthogonality of the complex exponential functions,

the $(n, k)^{th}$ element of \mathbf{W}^{-1} is $\frac{1}{N} \exp\left(-j\frac{2\pi}{N}nk\right)$.

$$\mathbf{W} \mathbf{W}^{-1} = \mathbf{W}^{-1} \mathbf{W} = \mathbf{I};$$

\mathbf{I} is the $N \times N$ identity matrix.

The columns of \mathbf{W} are linearly independent.



The eigenvalue relationship may be written as

$$\mathbf{W} \Lambda = \mathbf{C} \mathbf{W}, \quad (3.95)$$

where all the terms are $N \times N$ matrices, and

Λ is a diagonal matrix whose elements are equal to

$$\lambda(k), k = 0, 1, \dots, N - 1.$$

The expression above may be modified to

$$\mathbf{C} = \mathbf{W} \Lambda \mathbf{W}^{-1}. \quad (3.96)$$

Thus, a circulant matrix is diagonalized by the DFT operator \mathbf{W} .



Returning to periodic convolution, because \mathbf{h} is circulant,

$$\mathbf{h} = \mathbf{W} \mathbf{D}_h \mathbf{W}^{-1}, \quad (3.97)$$

where \mathbf{D}_h is a diagonal matrix (corresponding to Λ).

The elements of \mathbf{D}_h are given by multiplying the first row of the matrix \mathbf{h} in Equation 3.85 with W^{nk} , $n = 0, 1, 2, \dots, N - 1$, as in Equation 3.91:

$$\begin{aligned} H(k) &= h(0) + h(N-1)W^k + h(N-2)W^{2k} \\ &\quad + \dots + h(1)W^{(N-1)k}, \end{aligned} \quad (3.98)$$

which is a DFT relationship: $H(k)$ is the DFT of $h(n)$.



The series h above represents

$$h(N - n), n = 0, 1, 2, \dots, N - 1,$$

which is equal to $h(-n)$ due to periodicity.

The series of W values represents

$$\exp(+j\frac{2\pi}{N}nk), n = 0, 1, 2, \dots, N - 1.$$

The expression may be converted to the usual forward DFT form by substituting $-n = m$.



The result of the convolution operation is given by

$$\mathbf{g} = \mathbf{W} \mathbf{D}_h \mathbf{W}^{-1} \mathbf{f}. \quad (3.99)$$

$\mathbf{W}^{-1} \mathbf{f}$ is the (forward) DFT of \mathbf{f} (with a scale factor of $\frac{1}{N}$).

The multiplication of this expression by \mathbf{D}_h corresponds to point-by-point transform-domain filtering with the DFT of \mathbf{h} .

The multiplication by \mathbf{W} corresponds to the inverse DFT

(except for the scale factor $\frac{1}{N}$).



We now have the following equivalent relationships that represent convolution:

$$g(n) = h(n) * f(n)$$

$$G(k) = H(k) F(k)$$

$$\mathbf{g} = \mathbf{h} \mathbf{f}$$

$$\mathbf{g} = \mathbf{W} \mathbf{D}_h \mathbf{W}^{-1} \mathbf{f}. \quad (3.100)$$

Note: The representation of the Fourier transform operator above is different from that in Equation 3.67.



3.5.6 *Block-circulant matrix representation of a 2D filter*

Let \mathbf{f} represent the input image to a 2D filter.

Let \mathbf{g} represent the corresponding filtered image,

with the 2D arrays converted to vectors or column matrices.

Assume that all of the images have been padded with zeros and extended to $M \times N$ arrays, with M and N being large such that circular convolution is equivalent to linear convolution.

The images may be considered to be periodic (period = $M \times N$).

The matrices \mathbf{f} and \mathbf{g} are of size $MN \times 1$.



The 2D periodic convolution in array form is given by

$$g(m, n) = \sum_{\alpha=0}^{M-1} \sum_{\beta=0}^{N-1} f(\alpha, \beta) h([m-\alpha] \bmod M, [n-\beta] \bmod N), \quad (3.101)$$

for $m = 0, 1, 2, \dots, M - 1$, and $n = 0, 1, 2, \dots, N - 1$.

The result is also periodic with the period $M \times N$.



In order for $\mathbf{g} = \mathbf{h} \mathbf{f}$ to represent 2D convolution, we need to construct the matrix \mathbf{h} as follows:

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_0 & \mathbf{h}_{M-1} & \mathbf{h}_{M-2} & \cdots & \mathbf{h}_2 & \mathbf{h}_1 \\ \mathbf{h}_1 & \mathbf{h}_0 & \mathbf{h}_{M-1} & \cdots & \mathbf{h}_3 & \mathbf{h}_2 \\ \mathbf{h}_2 & \mathbf{h}_1 & \mathbf{h}_0 & \cdots & \mathbf{h}_4 & \mathbf{h}_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{h}_{M-1} & \mathbf{h}_{M-2} & \mathbf{h}_{M-3} & \cdots & \mathbf{h}_1 & \mathbf{h}_0 \end{bmatrix}. \quad (3.102)$$



The submatrices are given by

$$\mathbf{h}_m = \begin{bmatrix} h(m, 0) & h(m, N-1) & h(m, N-2) & \cdots & h(m, 1) \\ h(m, 1) & h(m, 0) & h(m, N-1) & \cdots & h(m, 2) \\ h(m, 2) & h(m, 1) & h(m, 0) & \cdots & h(m, 3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h(m, N-1) & h(m, N-2) & h(m, N-3) & \cdots & h(m, 0) \end{bmatrix}. \quad (3.103)$$

The matrix \mathbf{h} is of size $MN \times MN$.

Each $N \times N$ submatrix \mathbf{h}_m is a circulant matrix.

The submatrices of \mathbf{h} are subscripted in a circular manner:

\mathbf{h} is known as a *block-circulant* matrix.



Example: Consider filtering the image $f(m, n)$ given by

$$f(m, n) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 6 & 5 & 4 & 0 \\ 0 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The image has nonzero pixels over only a 3×3 region.

It has been padded with zeros to form a 5×5 array to allow for the result of convolution to be larger without wrap-around errors due to periodic convolution.



The 3×3 Laplacian operator, also extended to a 5×5 array, is given by

$$h(m, n) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -4 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

However, this form of the operator has its origin at the center of the array, whereas the origin of the image $f(m, n)$ as above would be at the top-left corner in matrix-indexing order.



Therefore, we need to rewrite $h(m, n)$ as follows:

$$h(m, n) = \begin{bmatrix} -4 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Observe that due to the assumption of periodicity, the values of $h(m, n)$ corresponding to negative indices now appear on the opposite ends of the matrix.



The matrices corresponding to the relationship $\mathbf{g} = \mathbf{h} \mathbf{f}$ are given in Figure 3.42. The resulting image $g(m, n)$ in array format is

$$g(m, n) = \begin{bmatrix} 0 & 1 & 2 & 3 & 0 \\ 1 & 4 & 1 & -6 & 3 \\ 6 & -11 & 0 & 1 & 4 \\ 7 & -14 & -11 & -24 & 9 \\ 0 & 7 & 8 & 9 & 0 \end{bmatrix}.$$



-4	1	0	0	1		1	0	0	0	0		0	0	0	0	0		0	0	0	0	0		1	0	0	0	0		0	0	0	0
1	-4	1	0	0		0	1	0	0	0		0	0	0	0	0		0	0	0	0	0		0	1	0	0	0		0	0	1	0
0	1	-4	1	0		0	0	1	0	0		0	0	0	0	0		0	0	0	0	0		0	0	1	0	0		0	2	0	0
0	0	1	-4	1		0	0	0	1	0		0	0	0	0	0		0	0	0	0	0		0	0	0	1	0		0	3	0	0
1	0	0	1	-4		0	0	0	0	1		0	0	0	0	0		0	0	0	0	0		0	0	0	0	1		0	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
1	0	0	0	0		-4	1	0	0	1		1	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0
0	1	0	0	0		1	-4	1	0	0		0	1	0	0	0		0	0	0	0	0		0	0	0	0	0		1	4	0	0
0	0	1	0	0		0	1	-4	1	0		0	0	1	0	0		0	0	0	0	0		0	0	0	0	0		2	1	1	0
0	0	0	1	0		0	0	1	-4	1		0	0	0	1	0		0	0	0	0	0		0	0	0	0	0		3	-6	0	0
0	0	0	0	1		1	0	0	1	-4		0	0	0	0	1		0	0	0	0	0		0	0	0	0	0		0	3	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0	0	0	0	0		1	0	0	0	0		-4	1	0	0	1		1	0	0	0	0		0	0	0	0	0		0	0	0	0
0	0	0	0	0		0	1	0	0	0		1	-4	1	0	0		0	1	0	0	0		0	0	0	0	0		6	0	0	0
0	0	0	0	0		0	1	0	0	0		0	1	-4	1	0		0	0	1	0	0		0	0	0	0	0		6	-11	0	0
0	0	0	0	0		0	0	1	0	0		0	1	-4	1	0		0	0	1	0	0		0	0	0	0	0		5	0	0	0
0	0	0	0	0		0	0	0	1	0		0	0	1	-4	1		0	0	0	1	0		0	0	0	0	0		4	1	0	0
0	0	0	0	0		0	0	0	0	1		1	0	0	1	-4		0	0	0	0	1		0	0	0	0	0		4	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0	0	0	0	0		0	0	0	0	0		1	0	0	0	0		-4	1	0	0	1		1	0	0	0	0		0	0	0	0
0	0	0	0	0		0	0	0	0	0		0	1	0	0	0		1	-4	1	0	0		0	1	0	0	0		7	-14	0	0
0	0	0	0	0		0	0	0	0	0		0	0	1	0	0		0	1	-4	1	0		0	0	1	0	0		8	-11	0	0
0	0	0	0	0		0	0	0	0	0		0	0	0	1	0		0	0	1	-4	1		0	0	0	1	0		9	-24	0	0
0	0	0	0	0		0	0	0	0	0		0	0	0	0	1		0	0	1	-4	1		0	0	0	0	1		0	9	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
1	0	0	0	0		0	0	0	0	0		0	0	0	0	0		1	0	0	0	0		-4	1	0	0	1		0	0	0	0
0	1	0	0	0		0	0	0	0	0		0	0	0	0	0		0	1	0	0	0		1	-4	1	0	0		0	7	0	0
0	0	1	0	0		0	0	0	0	0		0	0	0	0	0		0	0	1	0	0		0	1	-4	1	0		8	0	0	
0	0	0	1	0		0	0	0	0	0		0	0	0	0	0		0	0	0	1	0		0	0	1	-4	1		9	0	0	
0	0	0	0	1		0	0	0	0	0		0	0	0	0	0		0	0	0	0	1		1	0	0	1	-4		0	0	0	0

Figure 3.42: Matrices and vectors related to the application of the Laplacian operator to an image



Diagonalization of a block-circulant matrix:

Define the following functions related to the 2D DFT:

$$w_M(k, m) = \exp \left[j \frac{2\pi}{M} km \right], \text{ and } w_N(l, n) = \exp \left[j \frac{2\pi}{N} ln \right].$$

Define a matrix \mathbf{W} of size $MN \times MN$,

containing M^2 partitions each of size $N \times N$.



The $(k, m)^{th}$ partition of \mathbf{W} is

$$\mathbf{W}(k, m) = w_M(k, m) \ \mathbf{W}_N, \quad (3.104)$$

for $k, m = 0, 1, 2, \dots, M - 1$,

where \mathbf{W}_N is an $N \times N$ matrix with its elements given by

$w_N(l, n)$ for $l, n = 0, 1, 2, \dots, N - 1$.



Now, \mathbf{W}^{-1} is also a matrix of size $MN \times MN$,

with M^2 partitions of size $N \times N$.

The $(k, m)^{th}$ partition of \mathbf{W}^{-1} is

$$\mathbf{W}^{-1}(k, m) = \frac{1}{M} w_M^{-1}(k, m) \mathbf{W}_N^{-1}, \quad (3.105)$$

where $w_M^{-1}(k, m) = \exp[-j\frac{2\pi}{M}km]$,

for $k, m = 0, 1, 2, \dots, M - 1$.



The matrix \mathbf{W}_N^{-1} has its elements given by $\frac{1}{N}w_N^{-1}(l, n)$, where

$$w_N^{-1}(l, n) = \exp\left[-j\frac{2\pi}{N}ln\right], \text{ for } l, n = 0, 1, 2, \dots, N - 1.$$

Then, $\mathbf{W}\mathbf{W}^{-1} = \mathbf{W}^{-1}\mathbf{W} = \mathbf{I}$,

where \mathbf{I} is the $MN \times MN$ identity matrix.



If \mathbf{h} is a block-circulant matrix, it can be shown that

$$\mathbf{h} = \mathbf{W} \mathbf{D}_h \mathbf{W}^{-1}$$

$$\text{or } \mathbf{D}_h = \mathbf{W}^{-1} \mathbf{h} \mathbf{W},$$

where \mathbf{D}_h is a diagonal matrix whose elements are related to the DFT of $h(m, n)$, that is, to $H(k, l)$.



The following equivalent relationships represent 2D convolution:

$$g(m, n) = h(m, n) * f(m, n)$$

$$G(k, l) = H(k, l) F(k, l)$$

$$\mathbf{g} = \mathbf{h} \mathbf{f}$$

$$\mathbf{g} = \mathbf{W} \mathbf{D}_h \mathbf{W}^{-1} \mathbf{f}. \quad (3.106)$$



Note: Considering an $N \times N$ image $f(m, n)$, the

$N^2 \times N^2$ DFT matrix \mathbf{W} above is different from the

$N \times N$ DFT matrix \mathbf{W} in Equation 3.67.

The image matrix \mathbf{f} in Equation 3.67 is of size $N \times N$, whereas

the image is represented as an $N^2 \times 1$ vector in Equation 3.106.



Differentiation of functions of matrices:

Given the vectors \mathbf{f} and \mathbf{g} and a symmetric matrix \mathbf{W} ,

$$\frac{\partial}{\partial \mathbf{f}}(\mathbf{f}^T \mathbf{g}) = \frac{\partial}{\partial \mathbf{f}}(\mathbf{g}^T \mathbf{f}) = \mathbf{g}, \quad (3.107)$$

$$\frac{\partial}{\partial \mathbf{f}}(\mathbf{f}^T \mathbf{W} \mathbf{f}) = 2 \mathbf{W} \mathbf{f}. \quad (3.108)$$

Optimization of filters may be performed using matrix representation of images and image processing operations.



3.6 Optimal Filtering

3.6.1 The Wiener filter

Wiener filter: a linear filter designed to minimize the MSE between its output and the undegraded, unknown, original image.

The filter output is an optimal estimate of the original, undegraded image in the MSE sense: known as the

linear minimum mean squared-error (LMMSE) or the

least-mean-square (LMS) estimate.



Consider the degradation of an image \mathbf{f} by additive noise $\boldsymbol{\eta}$ that is independent of the image process.

We have the degraded image given by

$$\mathbf{g} = \mathbf{f} + \boldsymbol{\eta}. \quad (3.109)$$



Wiener estimation problem:

determine a linear estimate $\tilde{\mathbf{f}} = \mathbf{L}\mathbf{g}$ of \mathbf{f} from the given image \mathbf{g} , where \mathbf{L} is a linear filter or transform operator.

\mathbf{f} and \mathbf{g} are $N^2 \times 1$ matrices formed by row or column ordering of the corresponding $N \times N$ images.

\mathbf{L} is an $N^2 \times N^2$ matrix: the filter to be designed.

Optimization criterion: minimize the MSE

$$\varepsilon^2 = E \left[\|\mathbf{f} - \tilde{\mathbf{f}}\|^2 \right]. \quad (3.110)$$



MSE as the trace of the outer product matrix of the error vector:

$$\varepsilon^2 = E \left[\text{Tr} \left\{ (\mathbf{f} - \tilde{\mathbf{f}})(\mathbf{f} - \tilde{\mathbf{f}})^T \right\} \right]. \quad (3.111)$$

The following expressions are related to the above:

$$(\mathbf{f} - \tilde{\mathbf{f}})(\mathbf{f} - \tilde{\mathbf{f}})^T = \mathbf{f} \mathbf{f}^T - \mathbf{f} \tilde{\mathbf{f}}^T - \tilde{\mathbf{f}} \mathbf{f}^T + \tilde{\mathbf{f}} \tilde{\mathbf{f}}^T; \quad (3.112)$$

$$\tilde{\mathbf{f}}^T = \mathbf{g}^T \mathbf{L}^T = (\mathbf{f}^T + \boldsymbol{\eta}^T) \mathbf{L}^T; \quad (3.113)$$



$$\mathbf{f} \tilde{\mathbf{f}}^T = \mathbf{f} \mathbf{f}^T \mathbf{L}^T + \mathbf{f} \boldsymbol{\eta}^T \mathbf{L}^T; \quad (3.114)$$

$$\tilde{\mathbf{f}} \mathbf{f}^T = \mathbf{L} \mathbf{f} \mathbf{f}^T + \mathbf{L} \boldsymbol{\eta} \mathbf{f}^T; \quad (3.115)$$

$$\tilde{\mathbf{f}} \tilde{\mathbf{f}}^T = \mathbf{L} (\mathbf{f} \mathbf{f}^T + \mathbf{f} \boldsymbol{\eta}^T + \boldsymbol{\eta} \mathbf{f}^T + \boldsymbol{\eta} \boldsymbol{\eta}^T) \mathbf{L}^T. \quad (3.116)$$

The trace of a sum of matrices is equal to the sum of their traces:

E and Tr may be interchanged in order.



Applying the $E[]$ operator to the above, we get:

- $E[\mathbf{f} \ \mathbf{f}^T] = \boldsymbol{\phi}_f$, the autocorrelation matrix of the image;
- $E[\mathbf{f} \ \tilde{\mathbf{f}}^T] = \boldsymbol{\phi}_f \ \mathbf{L}^T$;
- $E[\mathbf{f} \ \boldsymbol{\eta}^T] = 0$ (\mathbf{f} and $\boldsymbol{\eta}$ assumed statistically independent) and $E[\boldsymbol{\eta}] = 0$;
- $E[\tilde{\mathbf{f}} \ \mathbf{f}^T] = \mathbf{L} \ \boldsymbol{\phi}_f$;
- $E[\tilde{\mathbf{f}} \ \tilde{\mathbf{f}}^T] = \mathbf{L} \ \boldsymbol{\phi}_f \ \mathbf{L}^T + \mathbf{L} \ \boldsymbol{\phi}_\eta \ \mathbf{L}^T$;
- $E[\boldsymbol{\eta} \ \boldsymbol{\eta}^T] = \boldsymbol{\phi}_\eta$, the noise autocorrelation matrix.



Now, the MSE may be written as

$$\begin{aligned}\varepsilon^2 &= \text{Tr} [\phi_f - \phi_f \mathbf{L}^T - \mathbf{L} \phi_f + \mathbf{L} \phi_f \mathbf{L}^T + \mathbf{L} \phi_\eta \mathbf{L}^T] \\ &= \text{Tr} [\phi_f - 2 \phi_f \mathbf{L}^T + \mathbf{L} \phi_f \mathbf{L}^T + \mathbf{L} \phi_\eta \mathbf{L}^T].\end{aligned}\quad (3.117)$$

$\text{Tr} [\phi_f \mathbf{L}^T] = \text{Tr} [\mathbf{L} \phi_f]$ because ϕ_f is symmetric.

The MSE is no longer a function of the images \mathbf{f} , \mathbf{g} , or $\boldsymbol{\eta}$:

depends only on the statistical characteristics of \mathbf{f} and $\boldsymbol{\eta}$, and \mathbf{L} .



To obtain the optimal filter operator \mathbf{L} :

differentiate the expression above with respect to \mathbf{L} ,
equate it to zero, and solve.

$$\frac{\partial \varepsilon^2}{\partial \mathbf{L}} = -2 \phi_f + 2 \mathbf{L} \phi_f + 2 \mathbf{L} \phi_\eta = 0. \quad (3.118)$$

$$\mathbf{L}_{Wiener} = \phi_f (\phi_f + \phi_\eta)^{-1}. \quad (3.119)$$

$$\tilde{\mathbf{f}} = \phi_f (\phi_f + \phi_\eta)^{-1} \mathbf{g}. \quad (3.120)$$



Implementation of the Wiener filter:

Consider the matrix $\phi_f + \phi_\eta$ that needs to be inverted.

The matrix would be of size $N^2 \times N^2$ for $N \times N$ images!

Inversion becomes easier if the matrix can be written as the product of a diagonal matrix and a unitary matrix.

ϕ_η is a diagonal matrix if η is uncorrelated random noise.



In most real images, correlation between pixels reduces as the spatial shift (distance) between the pixels considered increases:

Φ_f is then banded with several zeros, and may be approximated by a block-circulant matrix.

Then, $\Phi_f = \mathbf{W} \Phi_f \mathbf{W}^{-1}$ and

$\Phi_\eta = \mathbf{W} \Phi_\eta \mathbf{W}^{-1}$.



These steps lead to the Wiener filter output being expressed as

$$\tilde{\mathbf{f}} = \mathbf{W} \Phi_f (\Phi_f + \Phi_\eta)^{-1} \mathbf{W}^{-1} \mathbf{g}. \quad (3.121)$$

- $\mathbf{W}^{-1} \mathbf{g}$ equivalent to $G(k, l)$, Fourier transform of $g(m, n)$.
- $\Phi_f = \mathbf{W}^{-1} \phi_f \mathbf{W}$ equivalent to $S_f(k, l)$, PSD of $f(m, n)$.
- $\Phi_\eta = \mathbf{W}^{-1} \phi_\eta \mathbf{W}$ equivalent to $S_\eta(k, l)$, noise PSD.



We get the Wiener estimate in the Fourier domain as

$$\begin{aligned}\tilde{F}(k, l) &= \left[\frac{S_f(k, l)}{S_f(k, l) + S_\eta(k, l)} \right] G(k, l) \\ &= \left[\frac{1}{1 + \frac{S_\eta(k, l)}{S_f(k, l)}} \right] G(k, l).\end{aligned}\quad (3.122)$$

The Wiener filter transfer function depends upon the PSD of the original signal and noise processes;

the dependence is upon the second-order statistics of the processes rather than upon single realizations of the processes.



The gain of the Wiener filter varies from one frequency sample to another in accordance with the SNR

as a function of frequency in the 2D (u, v) or (k, l) space:

the gain is high wherever the signal component $S_f(k, l)$ is strong as compared to the noise component $S_\eta(k, l)$ (high SNR);

the gain is low wherever the SNR is low.

The gain is equal to unity if the noise PSD is zero.



The Wiener filter (as above) is not spatially adaptive: its characteristics remain the same for the entire image.

While suppressing noise, the Wiener filter is likely to blur sharp features and edges.



The design of appropriate signal and noise PSD models is difficult in practice;

inappropriate models could lead to poor performance.

Implicit assumption: the noise and image processes are second-order stationary processes;

mean and variance do not vary from one image region to another.

The entire image characterized by a single spectrum or PSD.

Most real-life images do not satisfy these assumptions:

need spatially or locally adaptive filters.



Example:

Figure 3.43 (a) shows the original Shapes test image.

Part (b): test image with Gaussian-distributed noise added;

$\mu = 0$, normalized $\sigma^2 = 0.01$.

Image PSD was modeled using a Laplacian function with $\sigma = 5$ pixels in the Fourier domain, represented using a 128×128 array.

Noise PSD was modeled by a uniform function having its total energy equal to 0.5 times that of the Laplacian PSD model.

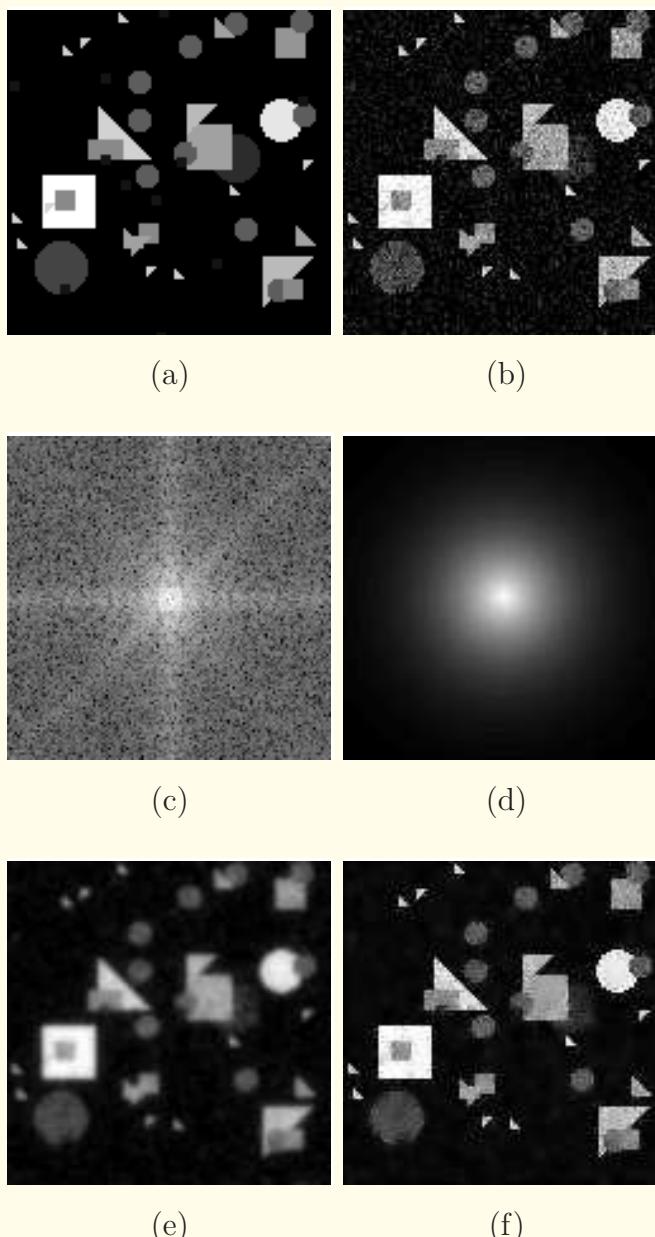


Figure 3.43: (a) Shapes test image. (b) Image in (a) with Gaussian-distributed noise added, with $\mu = 0$, normalized $\sigma^2 = 0.01$; RMS error = 19.56. (c) Log-magnitude spectrum of the image in (b). (d) Gain of the Wiener filter in the frequency domain, that is, the magnitude transfer function. Result of filtering the noisy image in (b) using: (e) the Wiener filter as in (d), RMS error = 52.89; (f) the local LMMSE filter with a 5×5 window, RMS error = 13.78.



3.7 Adaptive Filters

3.7.1 *The local LMMSE filter*

Adaptive, local-statistics-based filter to obtain the LMMSE estimate of the original image from a degraded version (Lee).

Degradation model:

$$g(m, n) = f(m, n) + \eta(m, n), \quad \forall m, n. \quad (3.123)$$



Data shown in the 2D array form with the indices (m, n) to indicate the pixel location in order to demonstrate the locally adaptive nature of the filter to be derived.

f : a realization of a *nonstationary random field*, characterized by spatially varying statistics.

Noise process η may be either signal-independent or signal-dependent, and could be nonstationary as well.



The LMMSE approach computes *at every spatial location* (m, n) an estimate $\tilde{f}(m, n)$ of the original image value $f(m, n)$ by applying a *linear* operator to $g(m, n)$.

Scalars $a(m, n)$ and $b(m, n)$ are sought such that

$$\tilde{f}(m, n) = a(m, n) g(m, n) + b(m, n) \quad (3.124)$$

minimizes the local MSE

$$\varepsilon^2(m, n) = \overline{[\tilde{f}(m, n) - f(m, n)]^2}. \quad (3.125)$$



The bar indicates some form of averaging
(statistical expectation, ensemble averaging, or spatial averaging).

Local MSE in expanded form:

$$\bar{\varepsilon}^2(m, n) = \overline{[a(m, n) g(m, n) + b(m, n) - f(m, n)]^2}. \quad (3.126)$$

Take the partial derivatives of $\bar{\varepsilon}^2(m, n)$ with respect to $a(m, n)$ and $b(m, n)$, set them to zero, and solve the resulting equations:



$$\frac{\partial \varepsilon^2(m, n)}{\partial b(m, n)} = \overline{2\{a(m, n) g(m, n) + b(m, n) - f(m, n)\}} = 0, \quad (3.127)$$

$$b(m, n) = \overline{f}(m, n) - a(m, n) \overline{g}(m, n). \quad (3.128)$$



Using $b(m, n)$ as above, we get the local MSE as

$$\varepsilon^2(m, n) = \overline{[a(m, n)\{g(m, n) - \bar{g}(m, n)\} - \{f(m, n) - \bar{f}(m, n)\}]^2}. \quad (3.129)$$

Differentiating with respect to $a(m, n)$ and setting to zero:

$$\overline{[a(m, n)\{g(m, n) - \bar{g}(m, n)\} - \{f(m, n) - \bar{f}(m, n)\}] \{g(m, n) - \bar{g}(m, n)\}} = 0. \quad (3.130)$$



Now, we may allow $\overline{[g(m, n) - \bar{g}(m, n)]^2} = \sigma_g^2(m, n)$

to represent the local variance of g , and

$$\overline{[g(m, n) - \bar{g}(m, n)][f(m, n) - \bar{f}(m, n)]} = \sigma_{fg}(m, n),$$

the local covariance between f and g . This leads to

$$a(m, n) = \frac{\sigma_{fg}(m, n)}{\sigma_g^2(m, n)}. \quad (3.131)$$



$$\tilde{f}(m, n) = \bar{f}(m, n) + \frac{\sigma_{fg}(m, n)}{\sigma_g^2(m, n)} [g(m, n) - \bar{g}(m, n)]. \quad (3.132)$$

The true statistics of the original and corrupted images as well as their joint statistics are unknown in a practical situation.

Lee proposed to estimate them locally in a spatial neighborhood of the pixel (m, n) being processed, leading to the

local LMMSE (that is, the LLMMSE) estimate.



Using a rectangular window of size $(2P + 1) \times (2Q + 1)$ centered at the pixel (m, n) being processed:

$$\mu_g(m, n) = \frac{1}{(2P + 1)(2Q + 1)} \sum_{p=-P}^P \sum_{q=-Q}^Q g(m+p, n+q), \quad (3.133)$$

$$\sigma_g^2(m, n) = \frac{1}{(2P + 1)(2Q + 1)} \sum_{p=-P}^P \sum_{q=-Q}^Q [g(m+p, n+q) - \mu_g(m, n)]^2. \quad (3.134)$$



Filter parameters are functions of space: space-variant entities.

LLMMSE estimate approximated by the pixel-by-pixel operation:

$$\tilde{f}(m, n) = \mu_g(m, n) + \left[\frac{\sigma_g^2(m, n) - \sigma_\eta^2(m, n)}{\sigma_g^2(m, n)} \right] [g(m, n) - \mu_g(m, n)]. \quad (3.135)$$

(also known as the Wiener filter; see Lim, Matlab)

The LLMMSE filter is rendered spatially adaptive and nonlinear by the space-variant estimation of the parameters.



Matrix representation:

$$\tilde{\mathbf{f}} = \mathbf{A}\mathbf{g} + \mathbf{b}. \quad (3.136)$$

$$\varepsilon^2 = E \left[Tr \left\{ (\mathbf{f} - \tilde{\mathbf{f}})(\mathbf{f} - \tilde{\mathbf{f}})^T \right\} \right]. \quad (3.137)$$



$$\begin{aligned}\varepsilon^2 &= E \left[\text{Tr} \left\{ (\mathbf{f} - \mathbf{A}\mathbf{g} - \mathbf{b})(\mathbf{f} - \mathbf{A}\mathbf{g} - \mathbf{b})^T \right\} \right] \\ &= E \left[\text{Tr} \left\{ \mathbf{f} \mathbf{f}^T - \mathbf{f} \mathbf{g}^T \mathbf{A}^T - \mathbf{f} \mathbf{b}^T - \mathbf{A} \mathbf{g} \mathbf{f}^T + \mathbf{A} \mathbf{g} \mathbf{g}^T \mathbf{A}^T \right. \right. \\ &\quad \left. \left. + \mathbf{A} \mathbf{g} \mathbf{b}^T - \mathbf{b} \mathbf{f}^T + \mathbf{b} \mathbf{g}^T \mathbf{A}^T + \mathbf{b} \mathbf{b}^T \right\} \right]. \quad (3.138)\end{aligned}$$



Differentiating with respect to \mathbf{b} and setting it to zero,

$$E [Tr \{-\mathbf{f} + \mathbf{Ag} - \bar{\mathbf{f}} + \mathbf{Ag} + 2\mathbf{b}\}] = 0, \quad (3.139)$$

solving which we get

$$\mathbf{b} = \bar{\mathbf{f}} - \mathbf{Ag}, \quad (3.140)$$

where $\bar{}$ indicates some form of averaging.



Using the expression derived for \mathbf{b} above, we get the following:

$$\begin{aligned}\mathbf{f} - \tilde{\mathbf{f}} &= \mathbf{f} - \mathbf{Ag} - \mathbf{b} \\ &= \mathbf{f} - \mathbf{Ag} - \bar{\mathbf{f}} + \mathbf{A}\bar{\mathbf{g}} \\ &= (\mathbf{f} - \bar{\mathbf{f}}) - \mathbf{A}(\mathbf{g} - \bar{\mathbf{g}}) \\ &= \mathbf{f}_1 - \mathbf{Ag}_1, \quad (3.141)\end{aligned}$$

$$\mathbf{f}_1 = \mathbf{f} - \bar{\mathbf{f}} \text{ and } \mathbf{g}_1 = \mathbf{g} - \bar{\mathbf{g}}$$

for the sake of compactness in further derivation.



$$\varepsilon^2 = E[Tr\{(\mathbf{f}_1 - \mathbf{A}\mathbf{g}_1)(\mathbf{f}_1 - \mathbf{A}\mathbf{g}_1)^T\}] \quad (3.142)$$

$$= E[Tr\{\mathbf{f}_1\mathbf{f}_1^T - \mathbf{f}_1\mathbf{g}_1^T\mathbf{A}^T - \mathbf{A}\mathbf{g}_1\mathbf{f}_1^T + \mathbf{A}\mathbf{g}_1\mathbf{g}_1^T\mathbf{A}^T\}].$$

Differentiating with respect to \mathbf{A} and setting to zero:

$$E[-2\mathbf{f}_1\mathbf{g}_1^T + 2\mathbf{A}\mathbf{g}_1\mathbf{g}_1^T] = 0. \quad (3.143)$$

$$E[\mathbf{g}_1\mathbf{g}_1^T] = E[(\mathbf{g} - \bar{\mathbf{g}})(\mathbf{g} - \bar{\mathbf{g}})^T] = \boldsymbol{\sigma}_g, \text{ covariance of } \mathbf{g}.$$

$$E[\mathbf{f}_1\mathbf{g}_1^T] = \boldsymbol{\sigma}_{fg}, \text{ the cross-covariance matrix of } \mathbf{f} \text{ and } \mathbf{g}.$$



Thus, we get $\mathbf{A} = \boldsymbol{\sigma}_{fg} \boldsymbol{\sigma}_g^{-1}$.

Finally, we obtain the LMMSE estimate as

$$\tilde{\mathbf{f}} = \bar{\mathbf{g}} + \boldsymbol{\sigma}_{fg} \boldsymbol{\sigma}_g^{-1} (\mathbf{g} - \bar{\mathbf{g}}). \quad (3.144)$$

This expression reduces to Equation 3.135 when local statistics are substituted for the expectation-based statistical parameters.

Example: see Figure 3.43 (f).



Refined LLMMSE filter:

If the local signal variance $\sigma_g^2(m, n)$ is high, it is assumed that the processing window is overlapping an edge.

The direction of the edge is computed using a gradient operator with eight possible directions for the edge.

According to the direction of the edge, the processing window is split into two sub-areas, each of which is assumed to be uniform.



The statistics computed within the sub-area that holds the pixel being processed are used.

Reduces noise in the neighborhood of edges without blurring.

Over uniform areas, $\sigma_g^2(m, n)$ is small; statistics computed over the whole processing window.

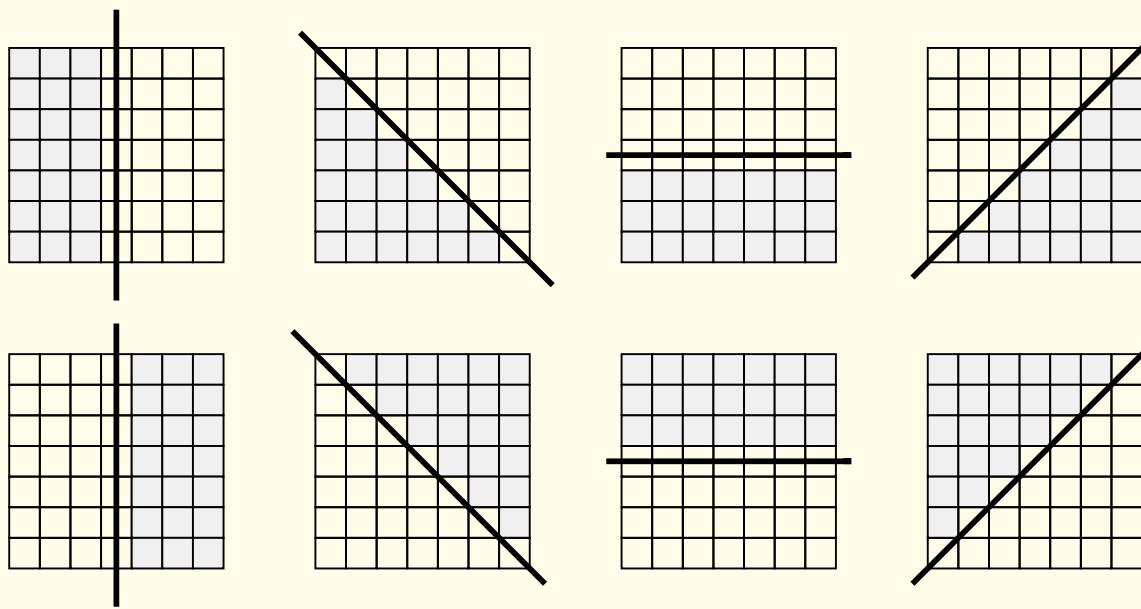


Figure 3.44: Splitting of a 7×7 neighborhood for adaptive filtering based upon the direction of a local edge detected within the neighborhood in a refined version of the local LMMSE filter. One of the eight cases shown is selected according to the direction of the gradient within the 7×7 neighborhood. Pixels in the partition containing the pixel being processed are used to compute the local statistics and the output of the filter. Based upon a similar figure in J.S. Lee, “Refined filtering of image noise using local statistics”, *Computer Graphics and Image Processing*, 15:380–389, 1981.



3.7.2 The noise-updating repeated Wiener filter

Deals with signal-independent additive, signal-dependent Poisson, and multiplicative noise (Jiang and Sawchuk).

Image: random field nonstationary in mean and variance.

Iterative application of the LLMMSE filter:

After each iteration, the variance of the noise is updated:

$$\begin{aligned} \sigma_{\eta}^{2new}(m, n) &= \left[1 - \frac{\sigma_{\eta}^2(m, n)}{\sigma_g^2(m, n)} + \frac{1}{(2P+1)(2Q+1)} \frac{\sigma_{\eta}^2(m, n)}{\sigma_g^2(m, n)} \right]^2 \sigma_{\eta}^2(m, n) \\ &+ \underbrace{\left[\frac{1}{(2P+1)(2Q+1)} \frac{\sigma_{\eta}^2(m, n)}{\sigma_g^2(m, n)} \right]^2}_{(p,q)\neq(0,0)} \sum_{p=-P}^{+P} \sum_{q=-Q}^{+Q} \sigma_{\eta}^2(m+p, n+q). \end{aligned} \quad (3.145)$$



3.7.3 The adaptive 2D LMS filter

A fixed-window Wiener filter in which the filter coefficients vary depending upon the image characteristics.

(Chan and Lim; Hadhoud and Thomas)

Based on the method of steepest descent;

tracks variations in the local statistics of the given image.

Does not require any *a priori* information about the image, the noise statistics, or their correlation properties.

No averaging, differentiation, or matrix operations.



2D LMS algorithm derived by defining a causal FIR filter $w_l(p, q)$ whose ROS is $P \times P$ (P typically being 3) such that

$$\tilde{f}(m, n) = \sum_{p=0}^{P-1} \sum_{q=0}^{P-1} w_l(p, q) g(m - p, n - q), \quad (3.146)$$

l marks the current position of the filter in the image,

$l = mM + n$ for the pixel position (m, n) in an $M \times N$ image,
 $l = 0$ to $MN - 1$.



Filter coefficients $w_{l+1}(p, q)$ for pixel $l + 1$ determined by minimizing the MSE between the desired pixel value $f(m, n)$ and the estimated pixel value $\tilde{f}(m, n)$ at the present pixel l , using the *method of steepest descent*.

$w_{l+1}(m, n)$ estimated as the present $w_l(p, q)$ plus a change proportional to the negative gradient of the error power (MSE):

$$w_{l+1}(p, q) = w_l(p, q) - \mu \nabla [e_l^2], \quad (3.147)$$

μ is a scalar controlling rate of convergence and stability;

e_l is the error signal;

∇ is a gradient operator [with respect to $w_l(p, q)$].



$$e_l = d(m, n) - \tilde{f}(m, n). \quad (3.148)$$

$d(m, n)$: approximation to original (unknown) image $f(m, n)$.

Hadhoud and Thomas obtained $d(m, n)$ from the input $g(m, n)$ by decorrelation, using the 2D delay operator of $(1, 1)$ samples.

$$w_{l+1}(p, q) = w_l(p, q) + 2 \mu e_l g(m - p, n - q), \quad (3.149)$$

a recursive equation defining the filter coefficients at the pixel position $l + 1$ in terms of those at the position l .

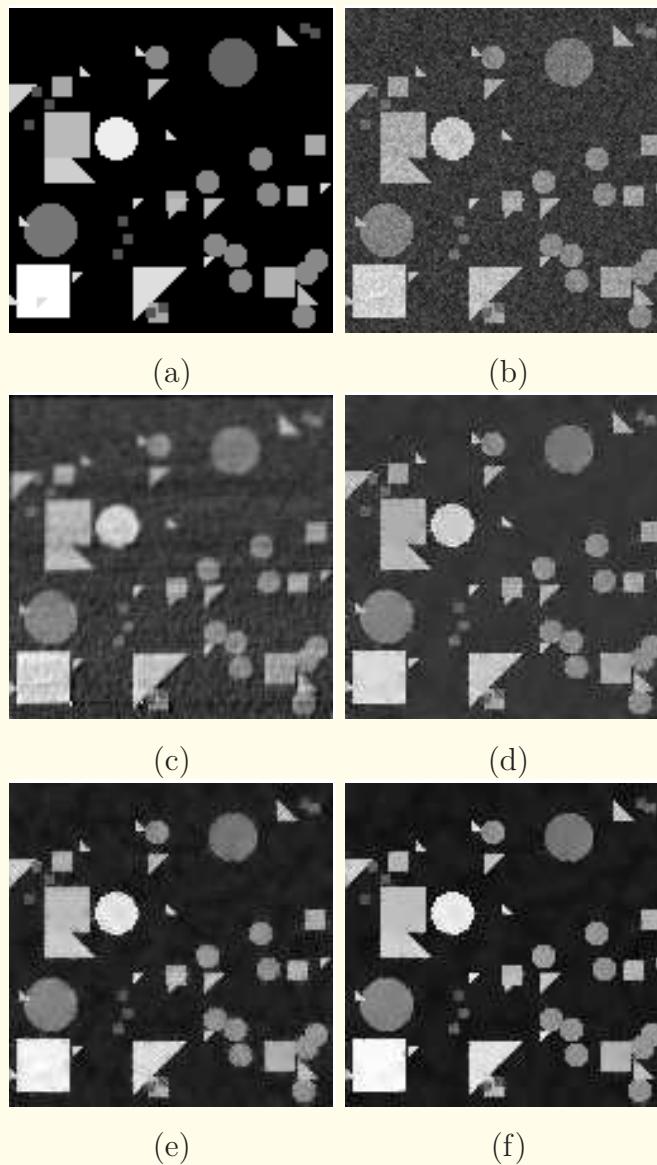


Figure 3.45: (a) “Shapes3”: a 128×128 test image with various geometrical objects placed at random. (b) Image in (a) with Gaussian noise added, RMS error = 14.24. Result of filtering the image in (b) with: (c) the 2D LMS filter, RMS error = 15.40; (d) two passes of the ARW-LMS filter, RMS error = 7.07; (e) the ANNS filter, RMS error = 6.68; (f) two passes of the ANNS filter, RMS error = 5.10. Reproduced with permission from R.B. Paranjape, T.F. Rabie, and R.M. Rangayyan, “Image restoration by adaptive-neighborhood noise subtraction”,



3.7.4 *The adaptive rectangular window LMS filter*

Proposed by Song and Pearlman; refined by Mahesh et al.

Same degradation model as in Equation 3.109;

additional assumption: image processes have zero mean.

$$\tilde{f}(m, n) = \alpha(m, n) g(m, n). \quad (3.150)$$



The problem reduces to finding $\alpha(m, n)$ at each pixel using the same minimum MSE criterion as that of the Wiener filter.

$$e(m, n) = f(m, n) - \tilde{f}(m, n) = f(m, n) - \alpha(m, n) g(m, n). \quad (3.151)$$

Minimization of the MSE requires that the error signal e be orthogonal to the image g , that is,

$$E\{[f(m, n) - \alpha(m, n) g(m, n)] g(m, n)\} = 0. \quad (3.152)$$



Solving for α , we obtain

$$\alpha(m, n) = \frac{\sigma_f^2(m, n)}{\sigma_f^2(m, n) + \sigma_\eta^2(m, n)}. \quad (3.153)$$

Because the noise is of zero mean, the *a posteriori* mean $\mu_g(m, n)$ is equal to the *a priori* mean $\mu_f(m, n)$; then

$$\tilde{f}(m, n) = \mu_g(m, n) \quad (3.154)$$

$$+ \frac{\sigma_f^2(m, n)}{\sigma_f^2(m, n) + \sigma_\eta^2(m, n)} [g(m, n) - \mu_g(m, n)].$$



Implementation of the ARW-LMS filter:

Assumption: pixels in the immediate neighborhood of a pixel (m, n) are samples from the same ensemble as that of $f(m, n)$;

a globally nonstationary process considered to be locally stationary and ergodic over a small region.

If we determine the size of a neighborhood in which the image values have the same statistical parameters, the sample statistics can approximate the *a posteriori* parameters in Equation 3.155.

Identify the size of a stationary rectangular region for each pixel; calculate local statistics of the image within that region.

Window size changes according to a measure of signal activity.



ARWs of odd length L_r in row and L_c in column direction.

$$L_r = 2N_r + 1 \text{ and } L_c = 2N_c + 1.$$

$$\tilde{\mu}_g(m, n) = \frac{1}{L_r L_c} \sum_{p=-N_r}^{+N_r} \sum_{q=-N_c}^{+N_c} g(m+p, n+q), \quad (3.155)$$

$$\tilde{\sigma}_g^2(m, n) = \frac{1}{L_r L_c} \sum_{p=-N_r}^{+N_r} \sum_{q=-N_c}^{+N_c} [g(m+p, n+q) - \tilde{\mu}_g(m, n)]^2. \quad (3.156)$$



$$\tilde{\sigma}_f^2 = \begin{cases} \tilde{\sigma}_g^2 - \sigma_\eta^2 & \text{if } \tilde{\sigma}_g^2 > \sigma_\eta^2 \\ 0 & \text{otherwise.} \end{cases} \quad (3.157)$$

$$\tilde{f}(m, n) = \tilde{\mu}_g(m, n) \quad (3.158)$$

$$+ \frac{\tilde{\sigma}_f^2(m, n)}{\tilde{\sigma}_f^2(m, n) + \sigma_\eta^2(m, n)} [g(m, n) - \tilde{\mu}_g(m, n)].$$

$L_r, L_c, N_r, N_c, \tilde{\mu}_g, \tilde{\sigma}_g^2$, and $\tilde{\sigma}_f^2$ and other parameters computed for each pixel (m, n) : $L_r(m, n)$, etc.

Noise variance σ_η^2 can be estimated from a window in a flat (signal-free) area of the degraded image.



L_r and L_c varied depending upon signal activity S_r :

$$S_r(m, n) = \frac{1}{L_r L_c} \sum_{p=-N_r}^{+N_r} \sum_{q=N_c}^{+N_c} [g(m + p, n + q) - \tilde{\mu}_r]^2 - \sigma_\eta^2, \quad (3.159)$$

where $\tilde{\mu}_r$ is the local mean evaluated in the row direction as

$$\tilde{\mu}_r = \frac{1}{L_r} \sum_{p=-N_r}^{+N_r} g(m + p, n). \quad (3.160)$$



S_r : measure of local roughness in row direction.

$$N_r \leftarrow N_r - 1, \text{ if } S_r \geq T_r, \quad (3.161)$$

$$\text{or } N_r \leftarrow N_r + 1, \text{ if } S_r < T_r. \quad (3.162)$$

$$T_r = \frac{\kappa \sigma_\eta^2}{L_r}, \quad (3.163)$$

κ : weighting factor to control rate of window size change.

Similar procedure in column direction to update N_c .

Example: in Figure 3.45 (d).



3.7.5 *The adaptive-neighborhood filter*

A variable-size, variable-shape neighborhood is determined individually for *every* pixel in the image.

The adaptive neighborhood grown for the pixel being processed (*the seed*) contains only those spatially connected pixels that are similar to the seed;

the neighborhood does not grow over edges but overlaps a stationary area.

The statistics computed using the pixels inside the adaptive region are likely to be close to the true statistics of the local signal and noise components.



Region growing for adaptive-neighborhood filtering:

A region needs to be grown for the pixel being processed (seed) such that it contains only pixels belonging to the same object or image feature as the seed.

The absolute difference between each of the 8-connected neighbors $g(p, q)$ and the seed $g(m, n)$ is computed as

$$d_{pq} = |g(p, q) - g(m, n)|. \quad (3.164)$$

Pixels $g(p, q)$ having $d_{pq} \leq T$, where T is a fixed, predefined threshold, are included in the region.



The procedure continues by checking the neighbors of the newly included pixels in the same manner, and stops when the inclusion criterion is not fulfilled for any neighboring pixel.

An adaptive neighborhood is grown for *each pixel* in the image.



In addition to the foreground region, an adaptive background region is also grown for each pixel.

The background is obtained by expanding (dilating) the outermost boundary of the foreground region by a prespecified number of pixels.

Redundant seed pixels: pixels within the foreground that have the same gray level as that of the seed pixel from where the process was commenced;

the use of a simple threshold for region growing will result in the same region being grown for all such pixels.

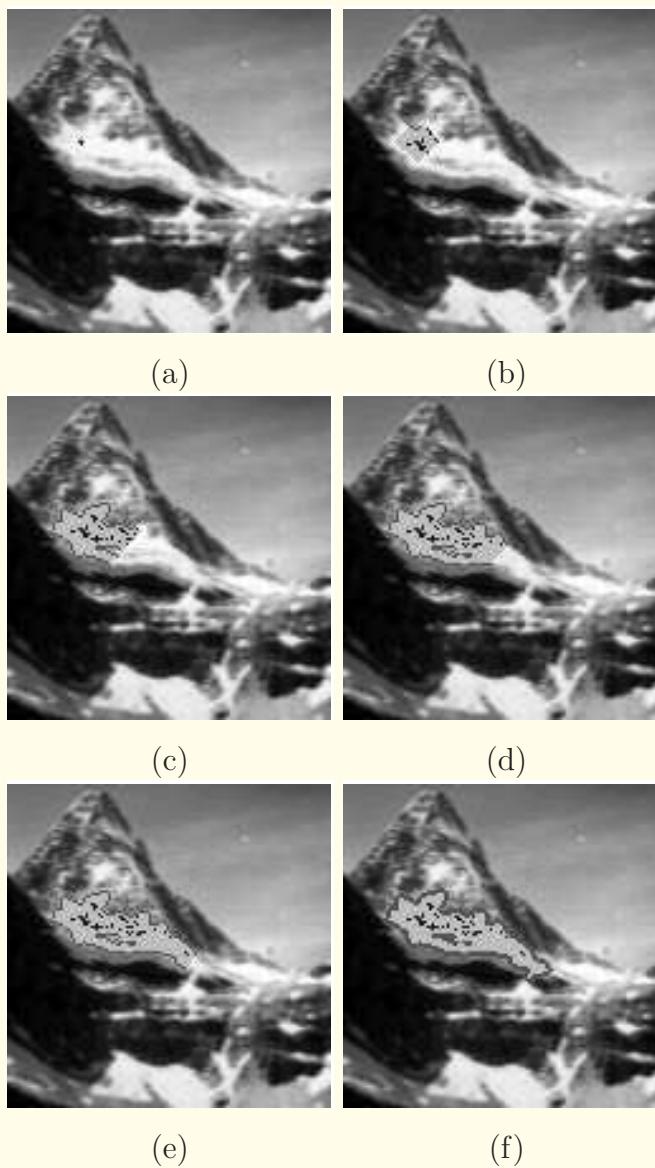


Figure 3.46: Illustration of the growth of an adaptive neighborhood (left to right and top to bottom). The neighborhood being grown is shown in a light shade of gray. The black pixels within the neighborhood have the same gray level as that of the seed from where the process was commenced: they are called redundant seed pixels. The last figure shows a region in a darker shade of gray that surrounds the foreground region: this is known as the adaptive background region. Figure courtesy of W.M. Morrow.

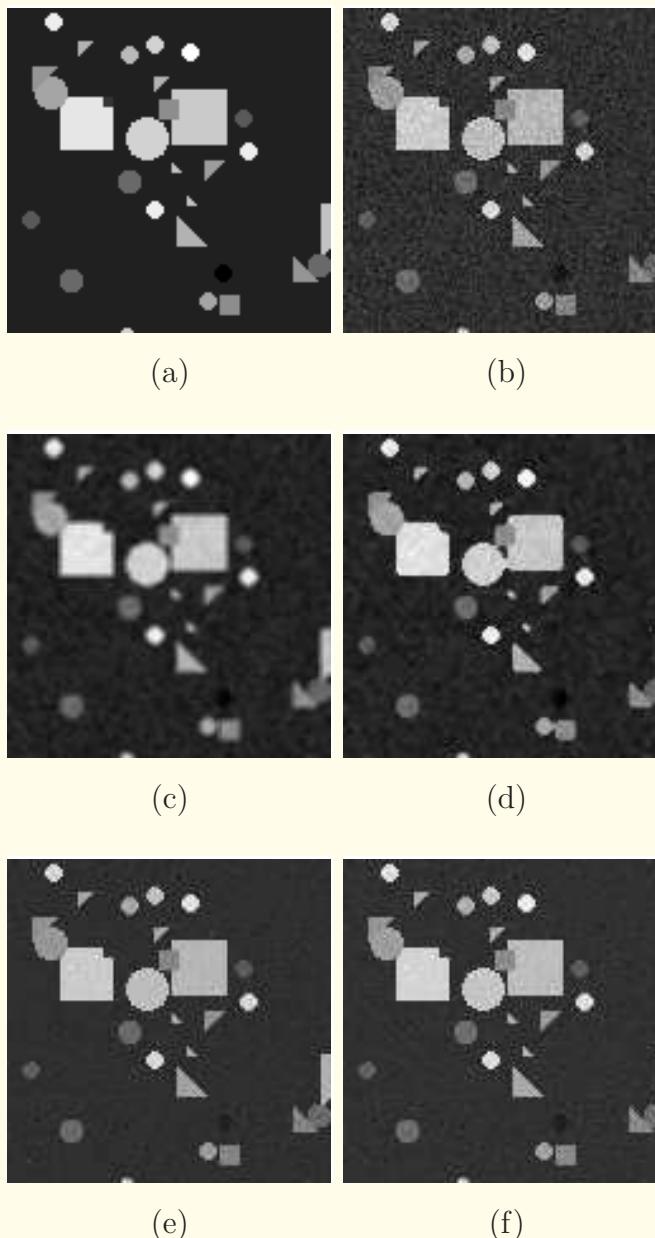


Figure 3.47: (a) “Shapes2”: a 128×128 test image with various geometrical objects placed at random. (b) Image in (a) with Gaussian noise added, RMS error = 8.24. Result of filtering the image in (b) with: (c) the 3×3 mean, RMS error = 9.24; (d) the 3×3 median, RMS error = 6.02; (e) the adaptive-neighborhood mean, RMS error = 3.16; (f) the adaptive-neighborhood median, RMS error = 4.01. Images courtesy of R.B. Paranjape.



Adaptive-neighborhood noise subtraction (ANNS):

The strategy used in deriving the ANNS filter is based upon the same principles as those of the ARW-LMS algorithm;

the image process f is assumed to be a zero-mean process of variance σ_f^2 that is observed in the presence of additive white Gaussian noise, resulting in the image g .

The noise process η is assumed to have zero mean and variance of σ_η^2 , and assumed to be uncorrelated to f .



An estimate of the additive noise at the pixel (m, n) is obtained from the corresponding adaptive neighborhood grown in the corrupted image g as follows:

$$\tilde{\eta}(m, n) = \alpha g(m, n), \quad (3.165)$$

where α is a scale factor which depends on the characteristics of the adaptive neighborhood grown.



Then, the estimate of $f(m, n)$ is

$$\tilde{f}(m, n) = g(m, n) - \tilde{\eta}(m, n), \quad (3.166)$$

which reduces to

$$\tilde{f}(m, n) = \beta g(m, n), \quad (3.167)$$

where $\beta = 1 - \alpha$.



As in the ARW-LMS algorithm, if the images used are of nonzero mean, the estimate of Equation 3.167 can be used by first subtracting the mean of each image.

$$\tilde{f}(m, n) = \mu_g(m, n) + (1 - \alpha) [g(m, n) - \mu_g(m, n)], \quad (3.168)$$

where μ_g is the *a posteriori* mean of the degraded image $g(m, n)$, which is also equal to the *a priori* mean μ_f of the original image $f(m, n)$ for zero-mean noise.



The problem now is to find the factor α , which is based upon the criterion that the estimated noise variance $\sigma_{\tilde{\eta}}^2$ be equal to the original noise variance σ_η^2 .

$$\sigma_\eta^2 = E[\tilde{\eta}^2]$$

$$= E \left[\{ \alpha [g(m, n) - \mu_g] \}^2 \right]$$

$$= \alpha^2 \sigma_g^2$$

$$= \alpha^2 (\sigma_f^2 + \sigma_\eta^2). \quad (3.169)$$



The noise estimation factor α is then given by

$$\alpha = \sqrt{\frac{\sigma_\eta^2}{\sigma_f^2 + \sigma_\eta^2}}. \quad (3.170)$$

Thus, the estimate of Equation 3.168 becomes

$$\tilde{f}(m, n) = \mu_g(m, n) \quad (3.171)$$

$$+ \left(1 - \sqrt{\frac{\sigma_\eta^2(m, n)}{\sigma_f^2(m, n) + \sigma_\eta^2(m, n)}} \right) [g(m, n) - \mu_g(m, n)].$$

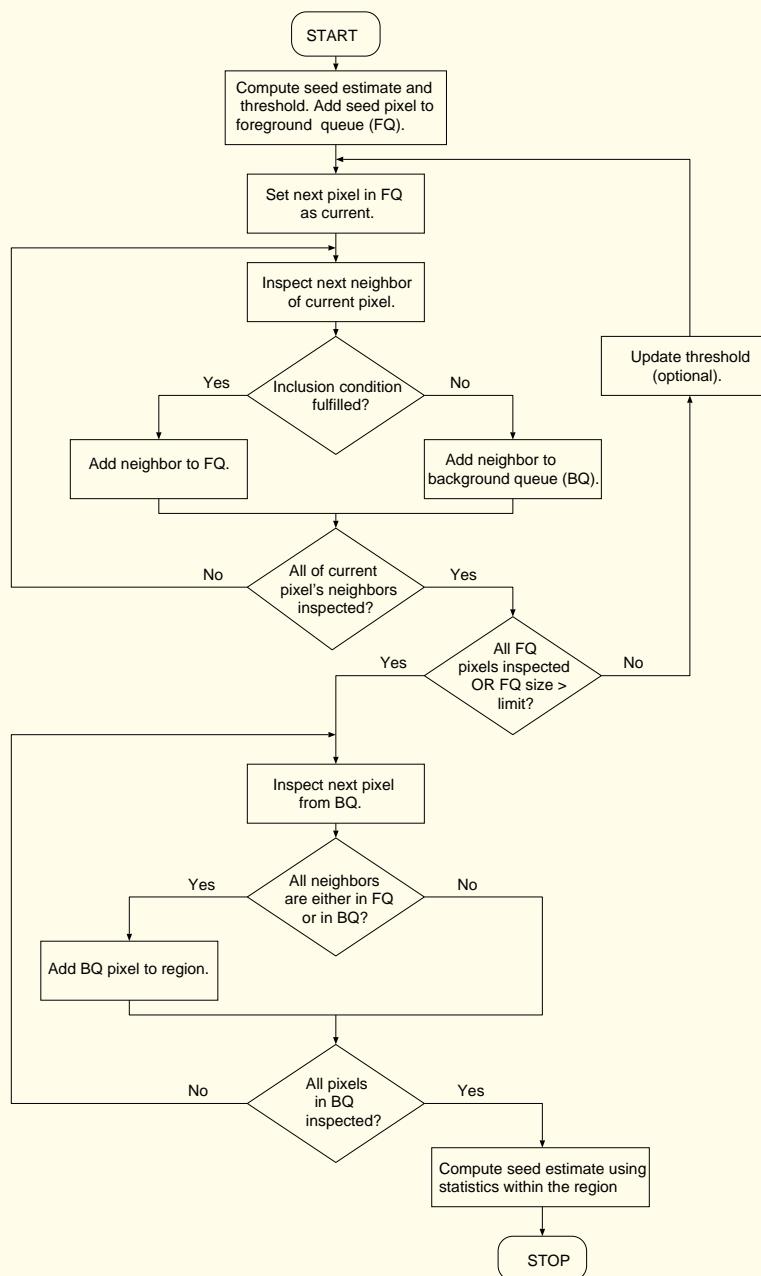


Figure 3.48: Flowchart of the adaptive region-growing procedure. Reproduced with permission from R.M. Rangayyan, M. Ciuc, and F. Faghah, “Adaptive neighborhood filtering of images corrupted by signal-dependent noise”, *Applied Optics*, 37(20):4477–4487, 1998. © Optical Society of America.

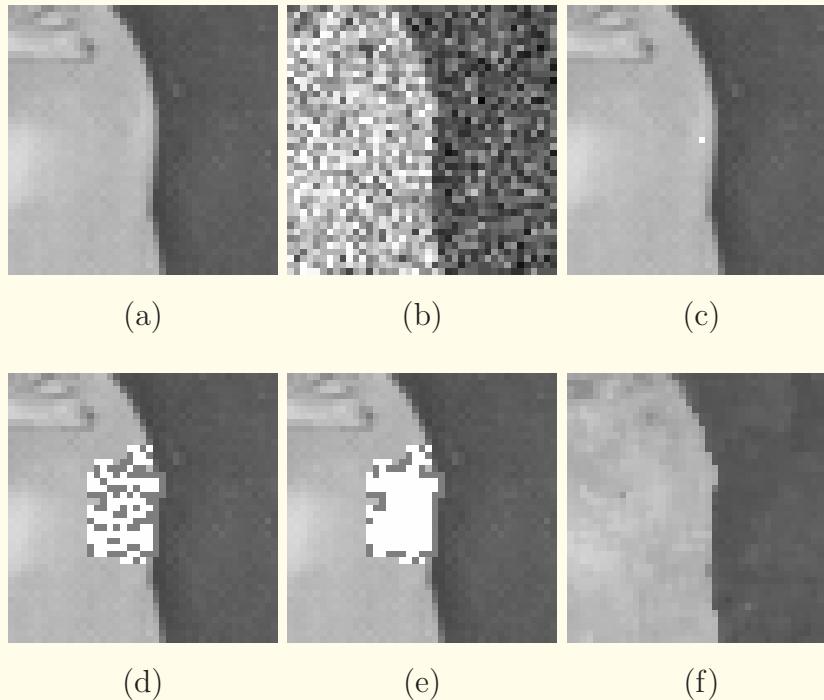


Figure 3.49: Illustration of the steps in adaptive region growing: (a) 25×25 pixel-wide portion of the original Peppers image. (b) Image corrupted by Poisson noise with $\lambda=0.1$. (c) The seed pixel, shown in white and located at the center of the image. (d) First step of region growing on the corrupted image: foreground pixels are in white, background pixels are in light gray. The foreground size has been limited to 100 pixels. (e) Region after inclusion of interior background pixels. (f) Filtered image. In (c), (d), and (e), the region has been superimposed over the uncorrupted image for convenience of display. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

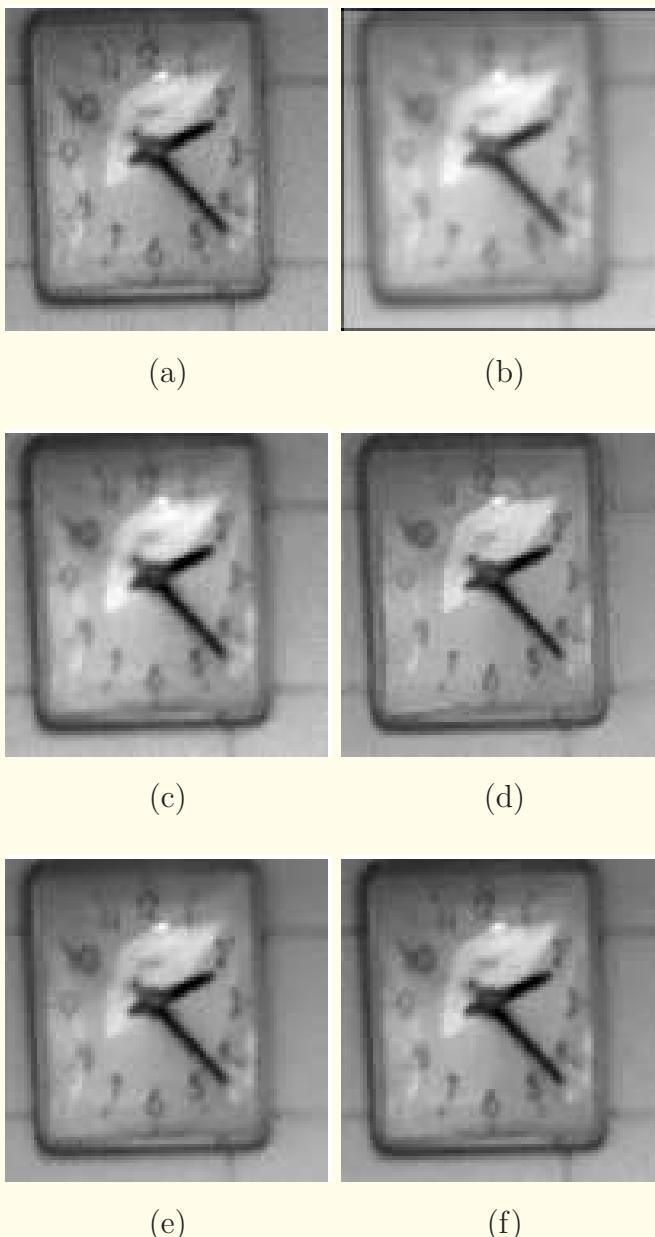


Figure 3.50: (a) Clock test image. Result of filtering the image in (a) using: (b) 3×3 mean; (c) 3×3 median; (d) the refined LLMMSE filter; (e) NURW filter; and (f) adaptive-neighborhood LLMMSE filter. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

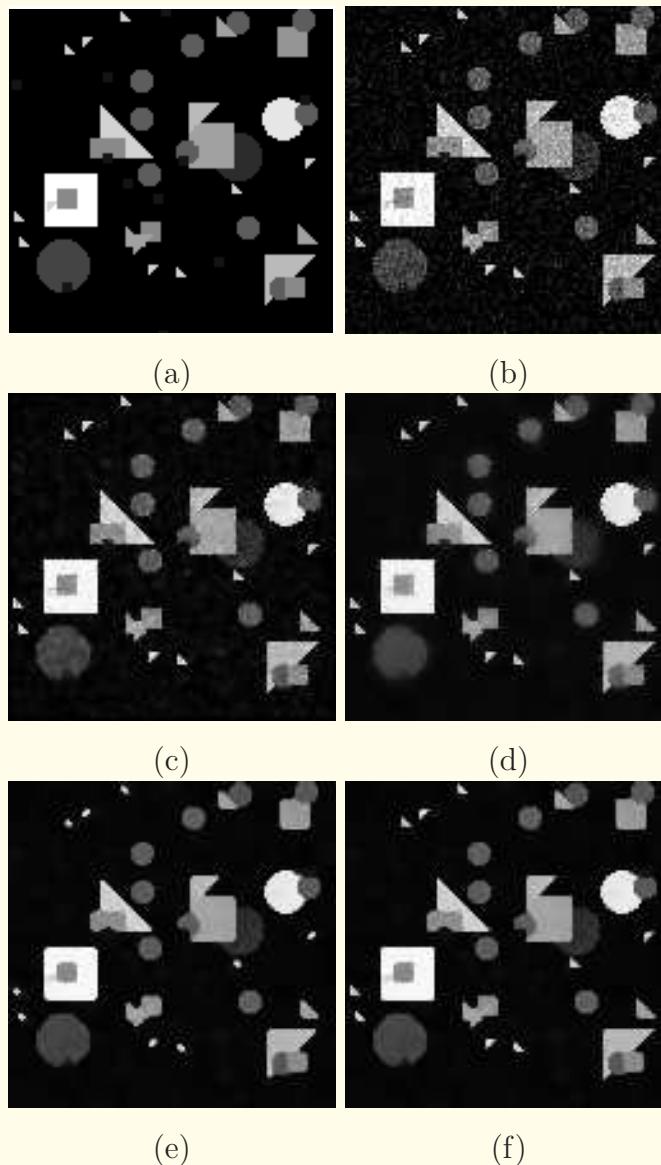


Figure 3.51: (a) Shapes: a 128×128 test image. (b) Image in (a) with Gaussian noise added, with $\mu = 0$, $\sigma = 20$, MSE = 228.75. Result of filtering the noisy image in (b) using: (c) 3×3 LLMMSE, MSE = 108.39; (d) NURW, MSE = 132.13; (e) adaptive-neighborhood mean, MSE = 205.04; and (f) adaptive-neighborhood LLMMSE, MSE = 78.58. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

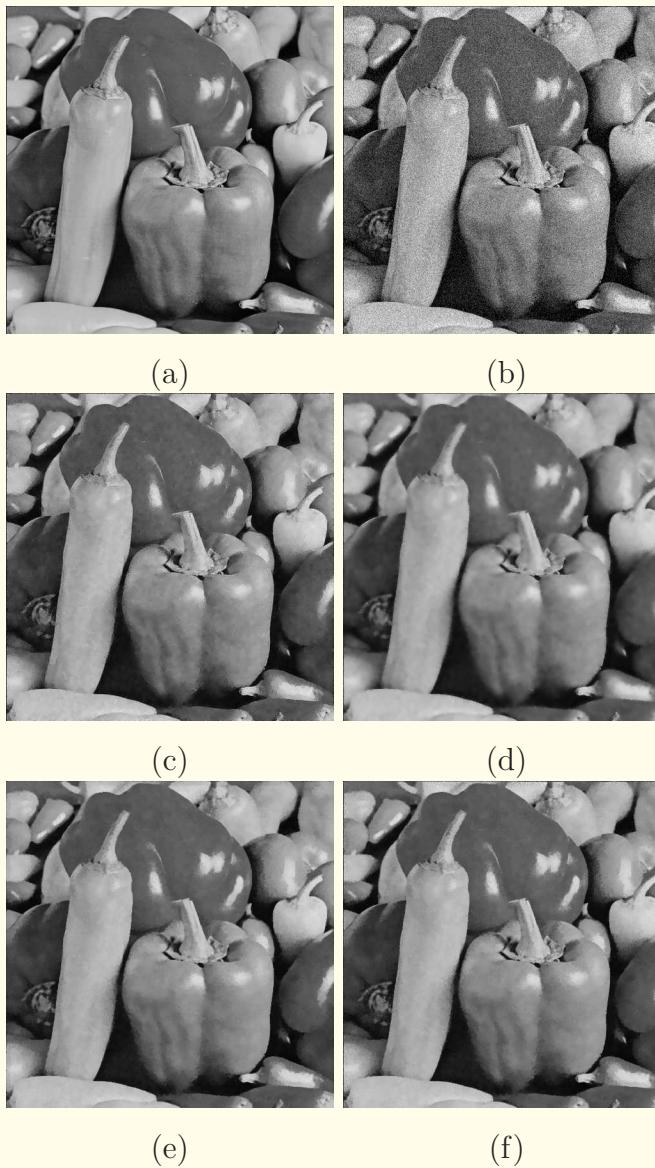


Figure 3.52: (a) 512×512 Peppers test image. (b) Image in (a) with Gaussian noise added, with $\mu = 0$, $\sigma = 20$, MSE = 389.87. Result of filtering the noisy image in (b) using: (c) refined LLMMSE, MSE = 69.49; (d) NURW, MSE = 54.70; (e) adaptive-neighborhood mean, MSE = 55.21; and (f) adaptive-neighborhood LLMMSE, MSE = 52.32. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.



Table 3.1: MSE values of the Noisy and Filtered Versions of the 128×128 Shapes Image.

TABLE 3.1

MSE values of the Noisy and Filtered Versions of the 128×128 Shapes Image.

Noise type	Noisy	3 Mean	3 Med.	5 Mean	5 Med.	3 LL	5 LL	R-LL	NURW	AN Mean	AN Med.	AN LL
Gaussian	228.75	469.26	213.71	772.76	518.17	108.39	122.29	124.88	132.13	205.04	197.93	78.58
Uniform	226.20	479.68	236.55	785.62	530.75	113.83	130.51	133.63	144.71	216.52	204.90	93.08
Poisson	241.07	441.04	266.85	743.73	657.35	108.70	130.14	131.47	147.87	215.18	249.41	62.57
Film-grain	275.11	450.92	283.74	746.90	665.78	119.81	147.27	141.64	166.42	236.27	296.81	69.98
Speckle	255.43	445.61	278.76	749.00	665.02	119.15	147.43	138.49	166.75	236.09	286.90	68.01
Salt & pepper	1740.86	642.20	206.63	835.46	557.75	1739.37	1405.09	1739.10	1740.84	213.02	205.72	1686.13

Note: 3 = 3×3 . 5 = 5×5 . Med. = Median. LL = LLMMSE. R = Refined. AN = Adaptive neighborhood.

Table 3.2: MSE Values of the Noisy and Filtered Versions of the 512×512 Peppers Image.

TABLE 3.2
MSE Values of the Noisy and Filtered Versions of the 512×512 Peppers Image.

Noise type	Noisy	3 Mean	3 Med.	5 Mean	5 Med.	3 LL	5 LL	R-LL	NURW	AN Mean	AN Med.	AN LL
Gaussian	389.87	74.89	93.55	84.88	71.80	86.53	68.19	69.49	54.70	55.21	57.50	52.32
Uniform	391.43	75.09	129.08	85.30	89.10	76.17	63.02	65.25	54.39	62.53	70.98	58.62
Poisson	1132.56	159.29	239.26	116.29	139.50	197.87	121.71	133.82	85.32	88.83	110.10	87.48
Film-grain	1233.43	168.25	245.77	119.59	135.32	212.03	125.25	117.54	88.83	90.54	101.19	89.07
Speckle	988.84	142.62	204.39	110.91	119.98	172.35	105.67	100.76	77.59	81.53	91.45	79.26
Salt & pepper	947.69	144.01	22.93	117.02	38.70	886.03	832.25	821.88	872.37	34.49	29.27	861.01

Note: 3 = 3×3 . 5 = 5×5 . Med. = Median. LL = LLMMSE. R = Refined. AN = Adaptive neighborhood.

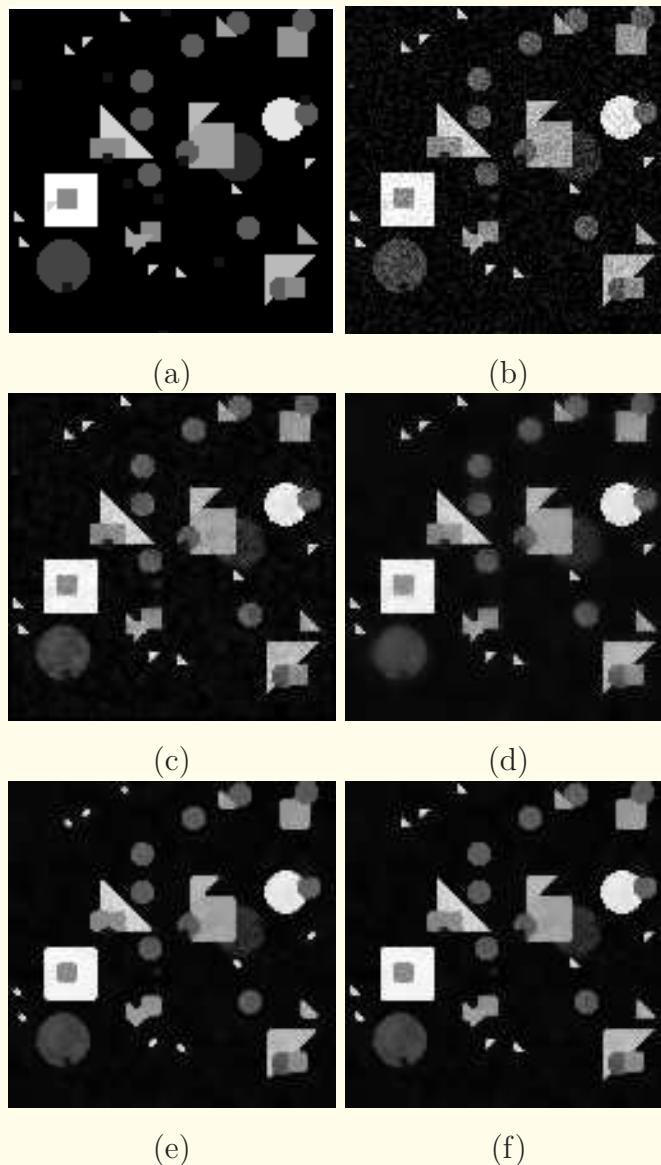


Figure 3.53: (a) Shapes test image. (b) Image in (a) with uniformly distributed noise added, with $\mu = 0$, $\sigma = 20$; MSE = 226.20. Result of filtering the noisy image in (b) using: (c) 3×3 LLMMSE; MSE = 113.83. (d) NURW; MSE = 144.71. (e) adaptive-neighborhood mean; MSE = 216.52. (f) adaptive-neighborhood LLMMSE; MSE = 93.08. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

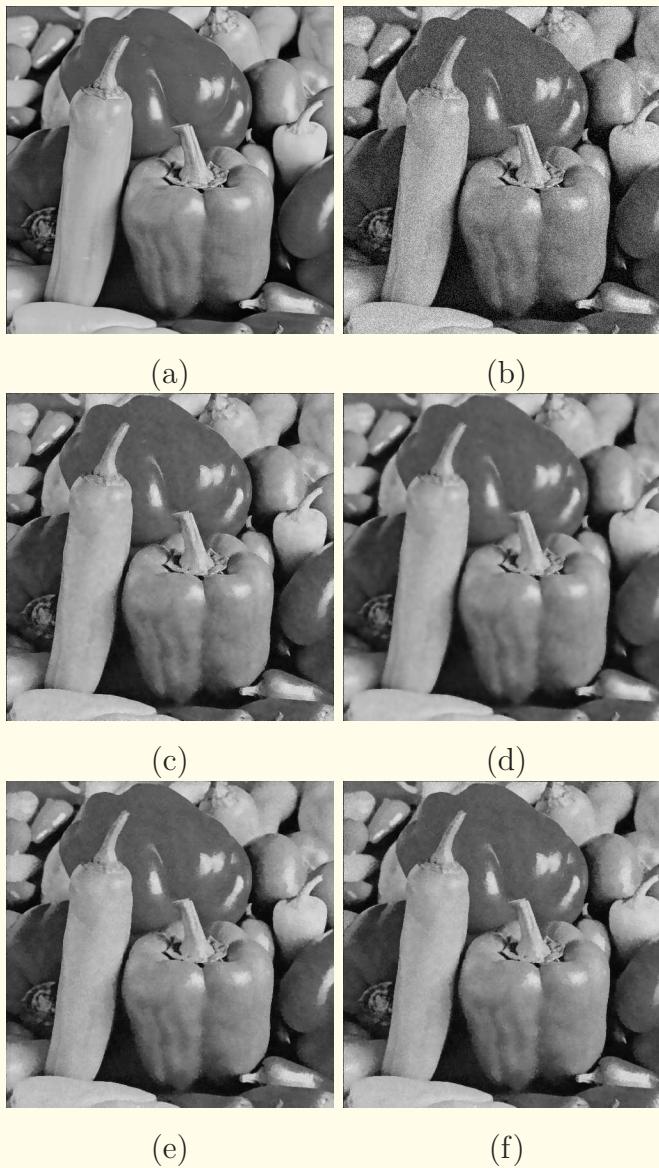


Figure 3.54: (a) Peppers test image. (b) Image in (a) with uniformly distributed noise added, with $\mu = 0$, $\sigma = 20$; MSE = 391.43. Result of filtering the noisy image in (b) using: (c) refined LLMMSE; MSE = 65.25. (d) NURW; MSE = 54.39. (e) adaptive-neighborhood mean; MSE = 62.53. (f) adaptive-neighborhood LLMMSE; MSE = 58.62. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

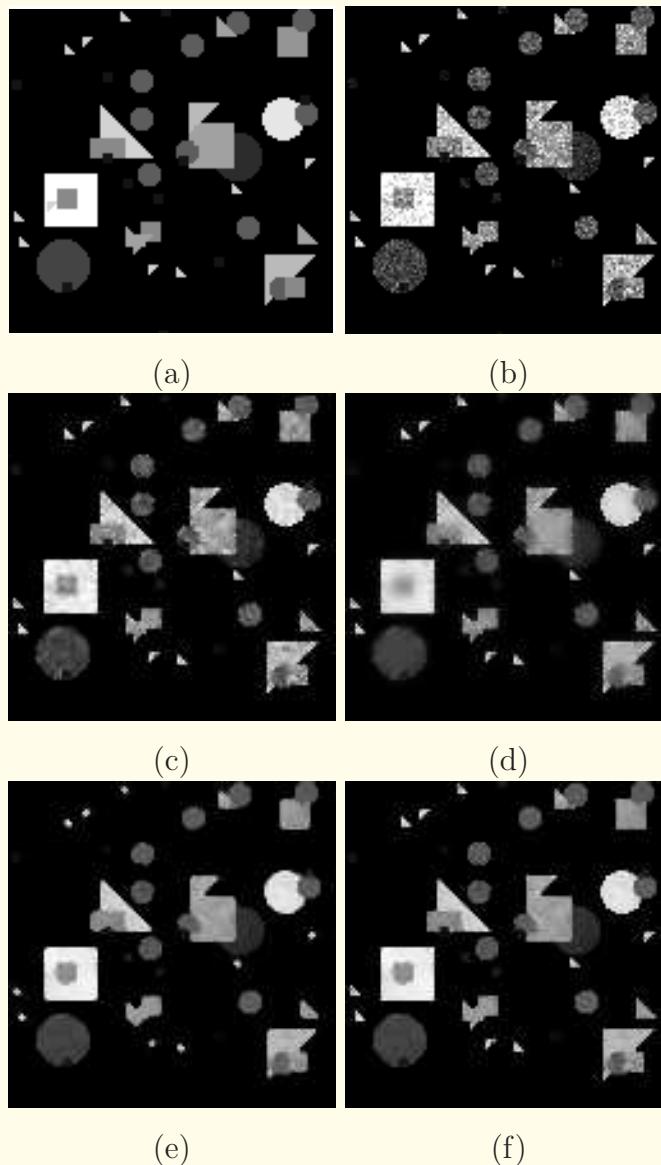


Figure 3.55: (a) Shapes test image. (b) Image in (a) with Poisson noise, with $\lambda = 0.1$. MSE = 241.07. Result of filtering the noisy image in (b) using: (c) 3×3 LLMMSE; MSE = 108.70. (d) NURW; MSE = 147.87. (e) adaptive-neighborhood mean; MSE = 215.18. (f) adaptive-neighborhood LLMMSE; MSE = 62.57. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imagineilor, Universitatea Politehnica București, Bucharest, Romania.

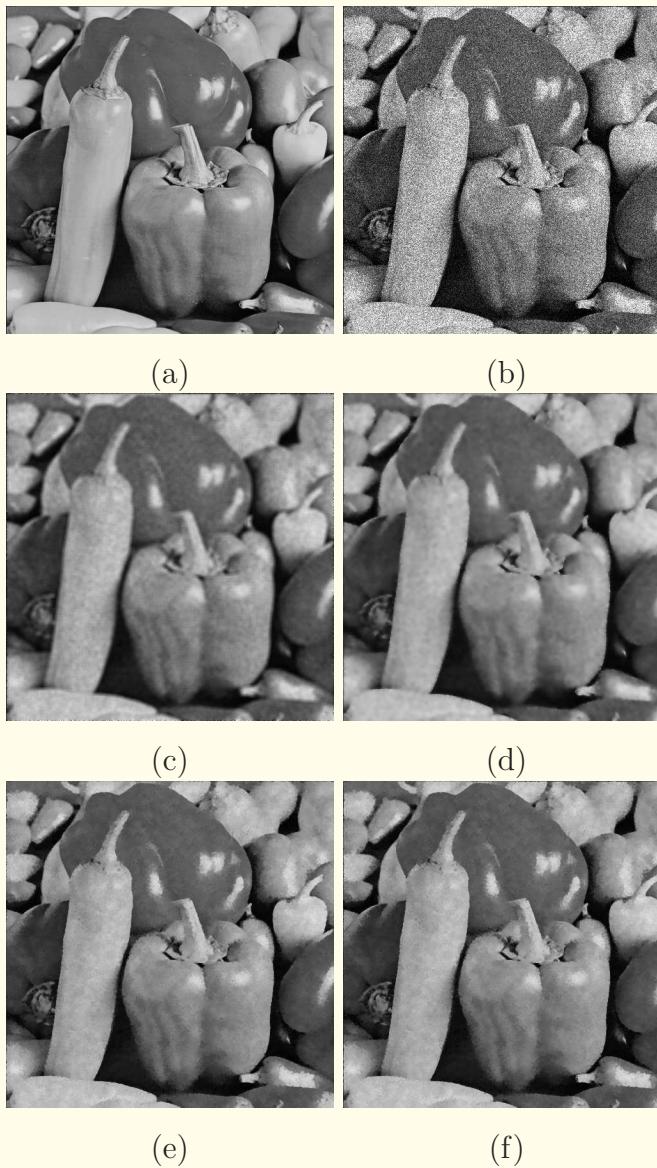


Figure 3.56: (a) Peppers test image. (b) Image in (a) with Poisson noise, with $\lambda = 0.1$. MSE = 1132.56. Result of filtering the noisy image in (b) using: (c) refined LLMMSE; MSE = 133.82. (d) NURW; MSE = 85.32. (e) adaptive-neighborhood mean; MSE = 88.83. (f) adaptive-neighborhood LLMMSE; MSE = 87.48. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imagineilor, Universitatea Politehnica București, Bucharest, Romania.

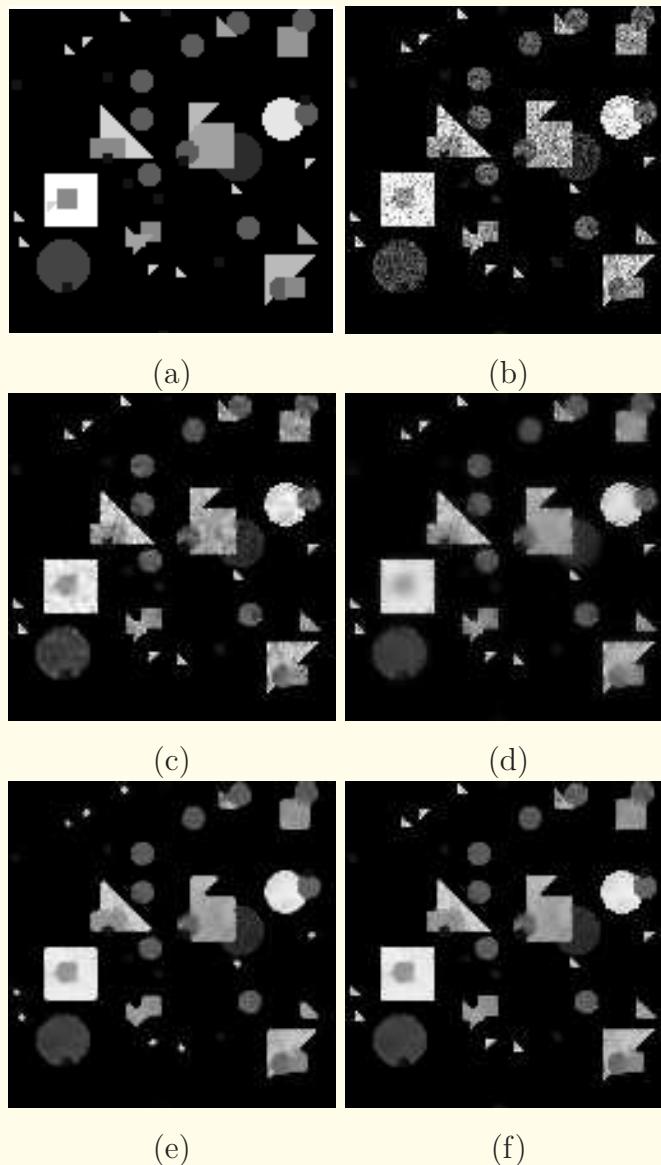


Figure 3.57: (a) Shapes test image. (b) Image in (a) with film-grain noise; MSE = 275.11. Result of filtering the noisy image in (b) using: (c) 3×3 LLMMSE; MSE = 119.81. (d) NURW; MSE = 166.42. (e) adaptive-neighborhood mean; MSE = 236.27. (f) adaptive-neighborhood LLMMSE; MSE = 69.98. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

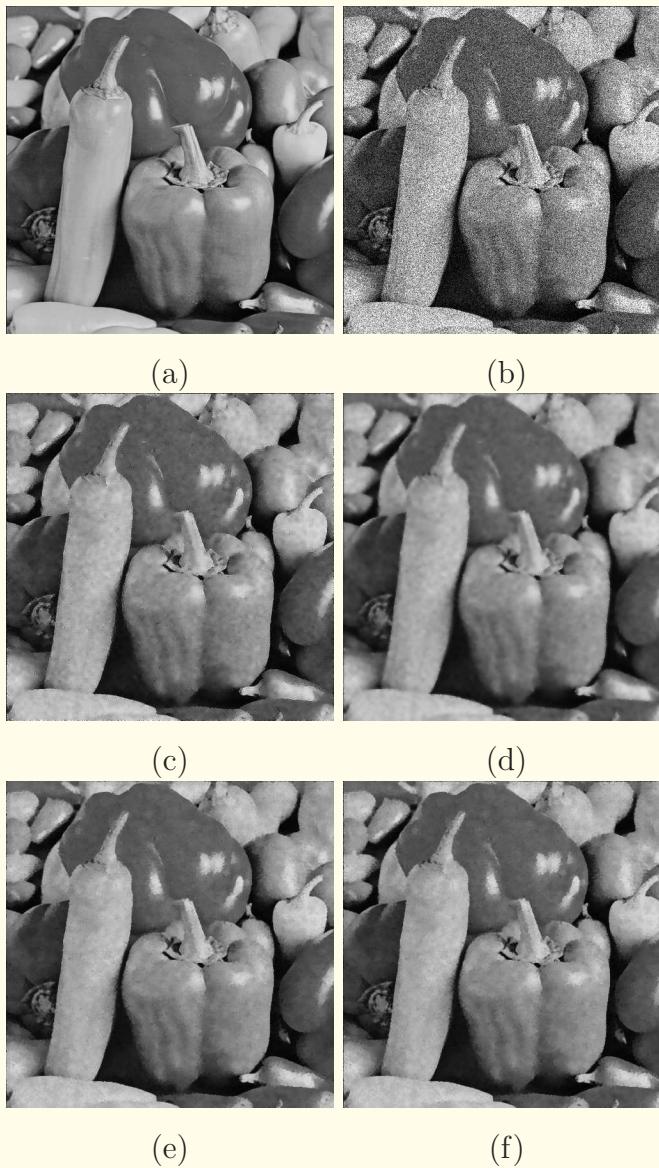


Figure 3.58: (a) Peppers test image. (b) Image in (a) with film-grain noise; MSE = 1233.43. Result of filtering the noisy image in (b) using: (c) refined LLMMSE; MSE = 117.54. (d) NURW; MSE = 88.83. (e) adaptive-neighborhood mean; MSE = 90.54. (f) adaptive-neighborhood LLMMSE; MSE = 89.07. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

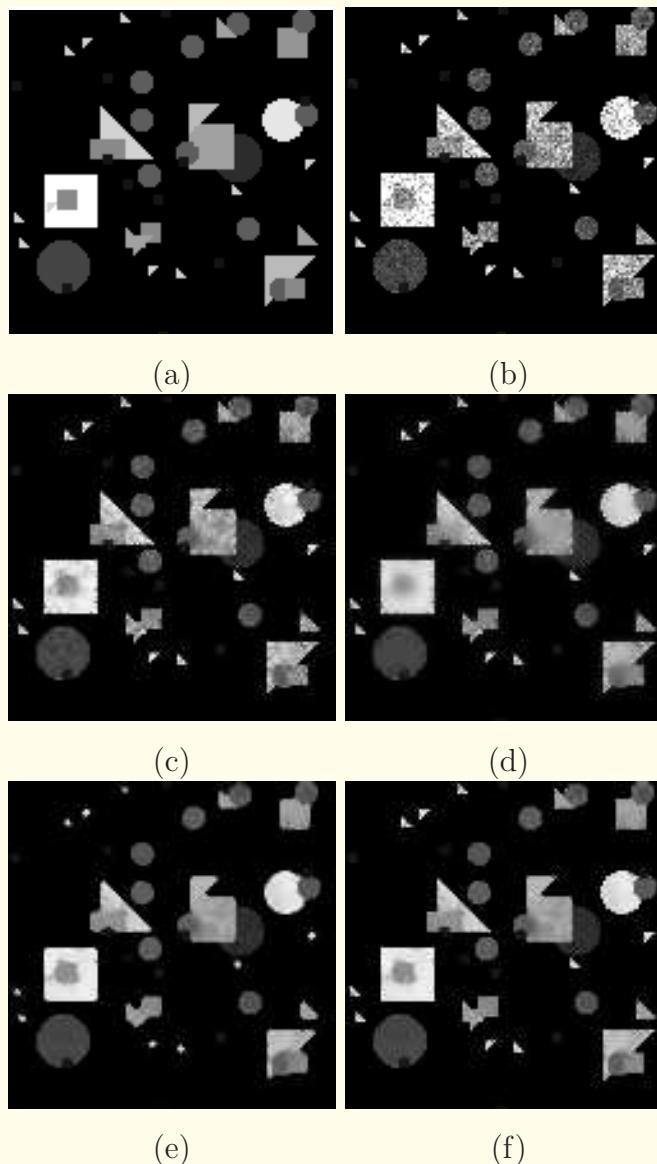


Figure 3.59: (a) Shapes test image. (b) Image in (a) with speckle noise, $MSE = 255.43$. Result of filtering the noisy image in (b) using: (c) 3×3 LLMMSE, $MSE = 119.15$; (d) NURW, $MSE = 116.75$; (e) adaptive-neighborhood mean, $MSE = 236.09$; (f) adaptive-neighborhood LLMMSE, $MSE = 68.01$. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

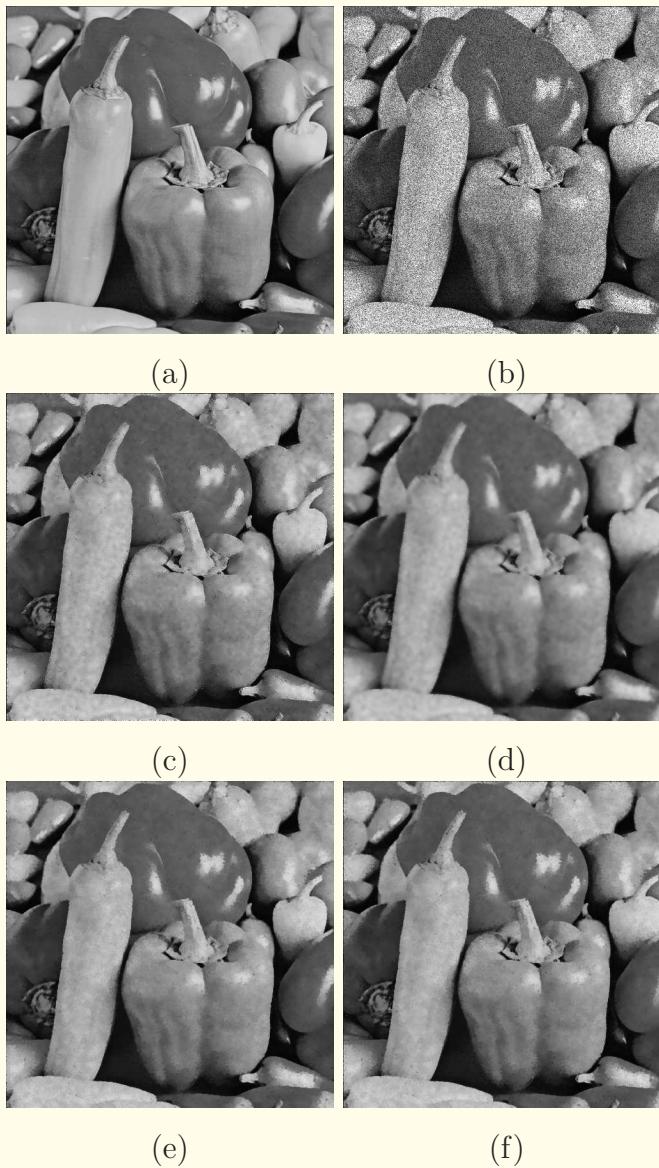


Figure 3.60: (a) Peppers test image. (b) Image in (a) with speckle noise, $MSE = 988.84$. Result of filtering the noisy image in (b) using: (c) refined LLMMSE, $MSE = 100.76$; (d) NURW, $MSE = 77.59$; (e) adaptive-neighborhood mean, $MSE = 81.54$; (f) adaptive-neighborhood LLMMSE, $MSE = 79.26$. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

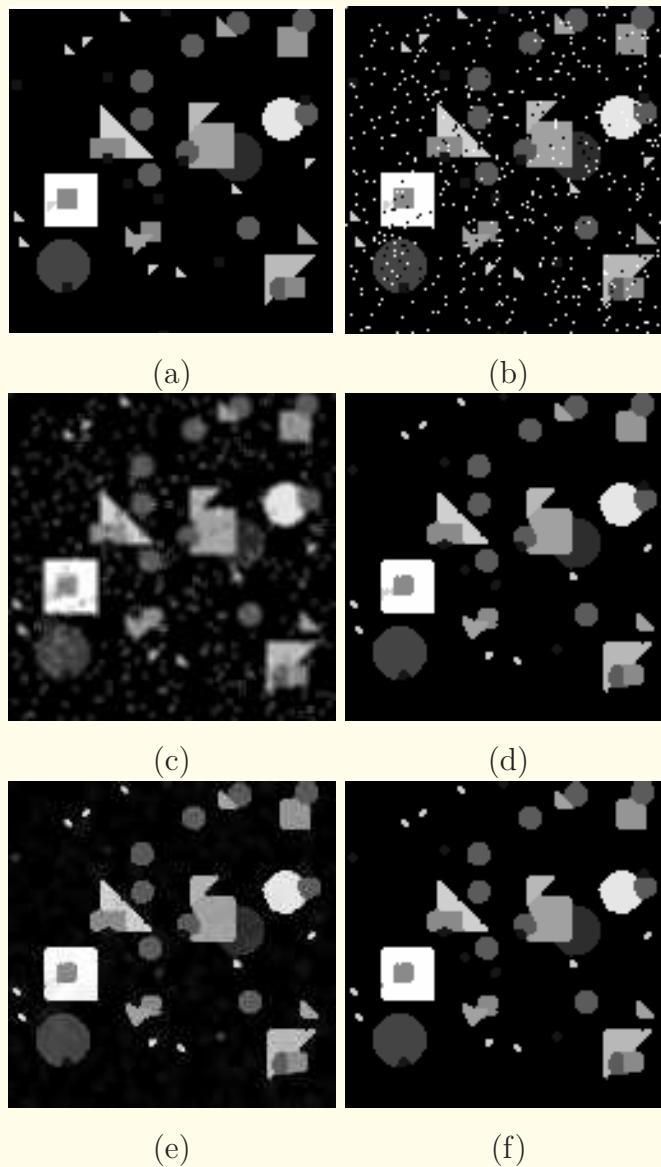


Figure 3.61: (a) Shapes test image. (b) Image in (a) with salt-and-pepper noise, MSE = 1740.86. Result of filtering the noisy image in (b) using: (c) 3×3 mean, MSE = 642.20; (d) 3×3 median, MSE = 206.63; (e) adaptive-neighborhood mean, MSE = 213.02; (f) adaptive-neighborhood median, MSE = 205.72. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

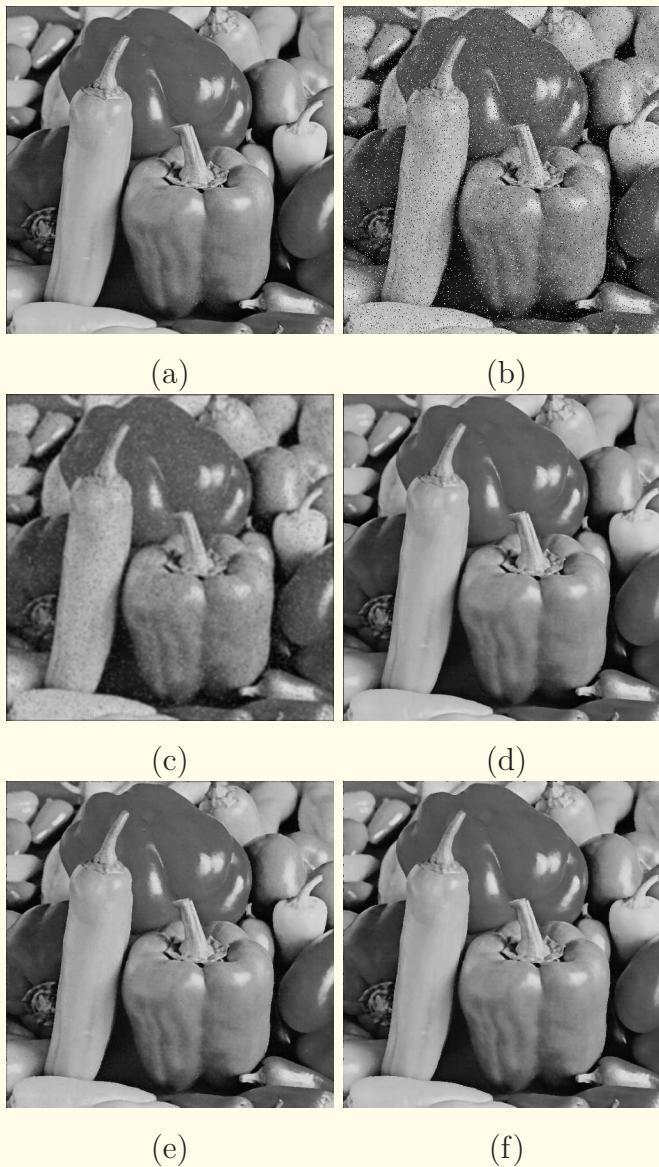


Figure 3.62: (a) Peppers test image. (b) Image in (a) with salt-and-pepper noise, MSE = 947.69. Result of filtering the noisy image in (b) using: (c) 5×5 mean, MSE = 117.02; (d) 3×3 median, MSE = 22.93; (e) adaptive-neighborhood mean, MSE = 34.49; (f) adaptive-neighborhood median, MSE = 29.27. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.



3.8 Application: Multiframe Averaging in Confocal Microscopy

The confocal microscope uses a laser beam to scan and image finely focused planes within fluorescent-dye-tagged specimens that could be a few *mm* in thickness.

Coherent light obviates the blur caused in imaging with ordinary white light, where the different frequency components of the incident light are reflected and refracted at different angles.

Laser excitation causes the dyes to emit light (fluoresce) at particular wavelengths.

The use of multiple dyes to stain different tissues and structures within the specimen permits their separate and distinct imaging.



The confocal microscope uses a pinhole to permit the passage of only the light from the plane of focus; light from the other planes of the specimen is blocked.

A PMT is used to amplify the light received.

A scanning mechanism is used to raster-scan the sample in steps that could be as small as $0.1 \mu m$.

The confocal microscope facilitates the imaging of multiple focal planes separated by distances of the order of $1 \mu m$;

several such slices may be acquired and combined to build 3D images of the specimen.



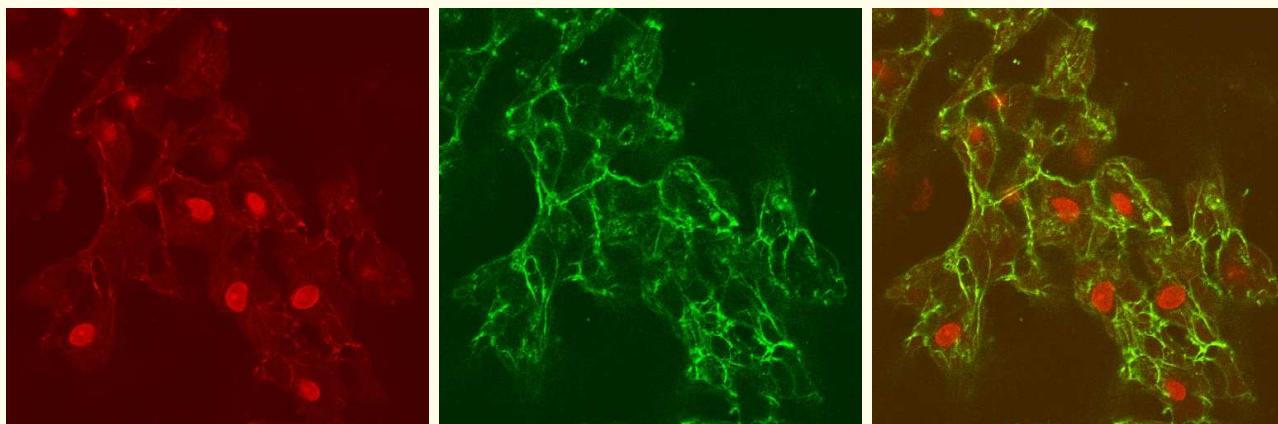
Example: Figure 3.63 shows images of cells from the nucleus pulposus (the central portion of the intervertebral discs), scanned using a laser beam of wavelength 488 nm .

Red-dye (long-pass cutoff at 585 nm): cell nuclei.

Green-dye (pass band of $505 - 530\text{ nm}$): actin filament structure.

Focal plane thickness about $1\text{ }\mu\text{m}$.

Multiframe averaging used to reduce noise.



(a)

(b)

(c)

Figure 3.63: (a) The red-dye (cell nuclei) component of the confocal microscope image of the nucleus pulposus of a dog. (b) The green-dye (actin filament structure) component. (c) Combination of the images in (a) and (b) into a composite image. The images would be viewed in the colors mentioned on the microscope. The width of each image corresponds to $145 \mu\text{m}$. Each image was acquired by averaging eight frames. Images courtesy of C.J. Hunter, J.R. Matyas, and N.A. Duncan, McCaig Centre for Joint Injury and Arthritis Research, University of Calgary.

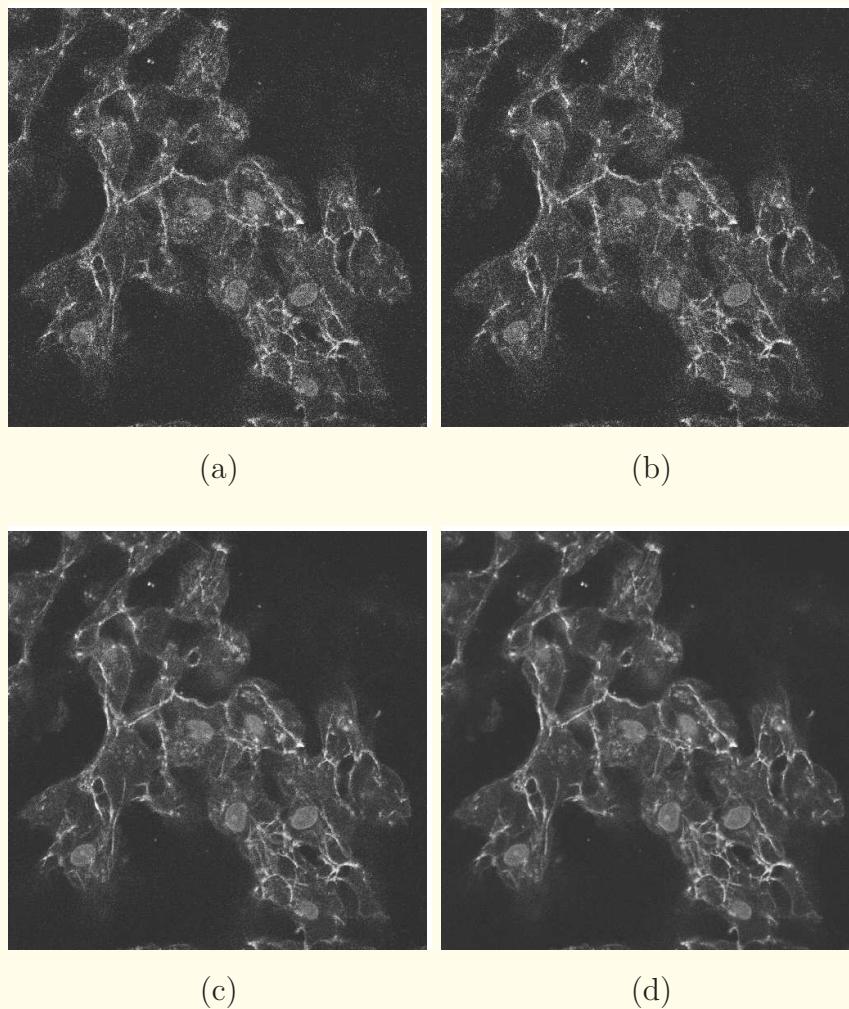


Figure 3.64: (a) A single-frame acquisition of the composite image of the nucleus pulposus; see also Figure 3.63. (b) A second example of a single-frame acquisition as in (a). (c) The result of averaging four frames including the two in (a) and (b). (d) The result of averaging eight frames including the two in (a) and (b). The width of each image corresponds to [145 \$\mu\text{m}\$](#) . Images courtesy of C.J. Hunter, J.R. Matyas, and N.A. Duncan, McCaig Centre for Joint Injury and Arthritis Research, University of Calgary.

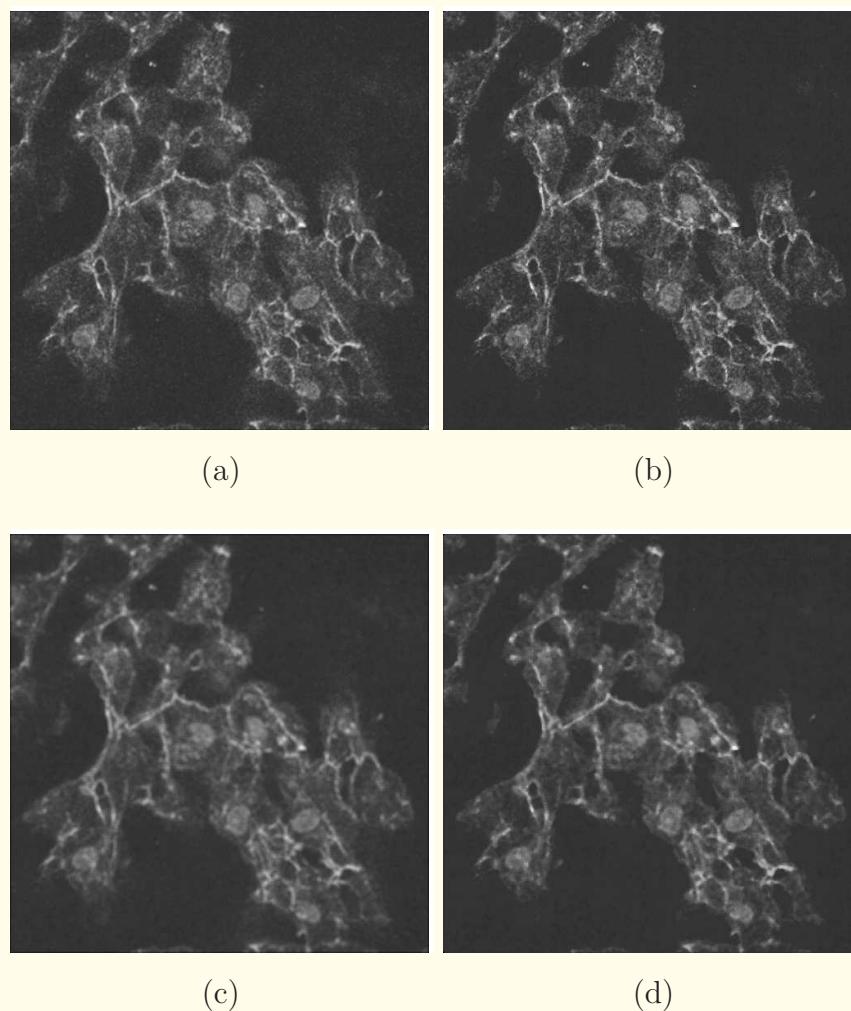


Figure 3.65: Results of filtering the single-frame acquisition of the composite image of the nucleus pulposus in Figure 3.64 (a) with: (a) the 3×3 mean filter; (b) the 3×3 median filter; (c) the 5×5 mean filter; and (d) the 5×5 median filter.

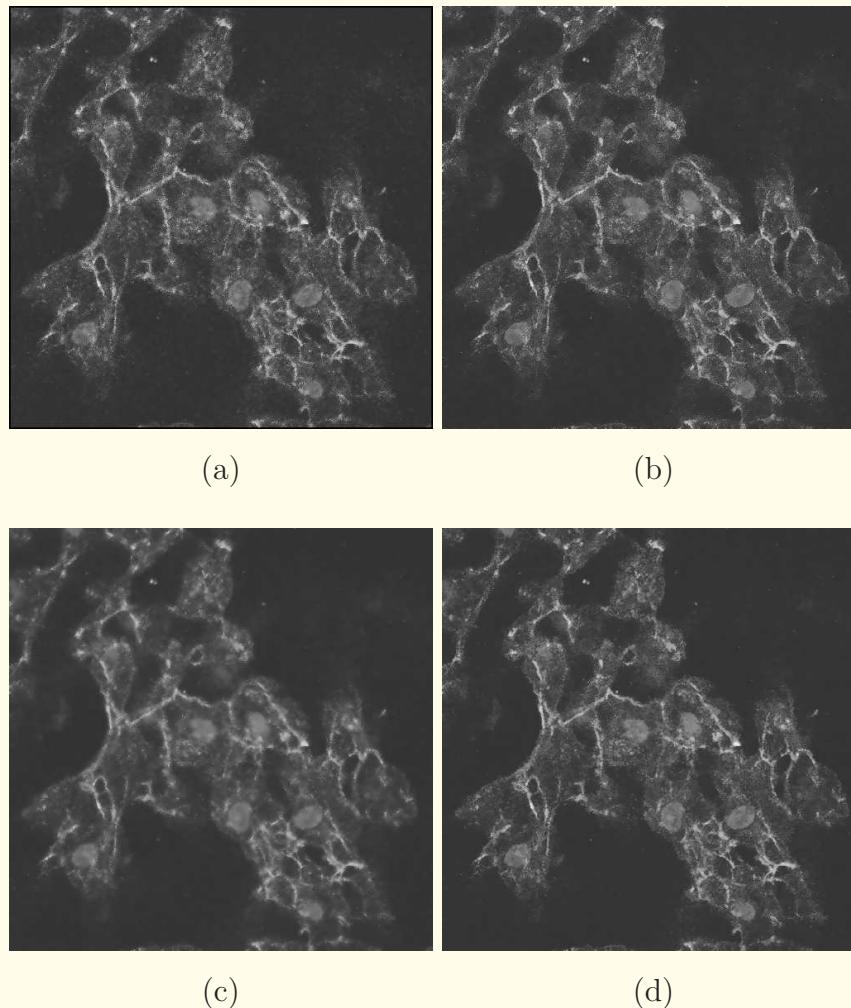


Figure 3.66: Results of filtering the single-frame acquisition of the composite image of the nucleus pulposus in Figure 3.64 (a) with: (a) the 5×5 LLMMSE filter; (b) the refined LLMMSE filter; (c) the NURW filter; and (d) the adaptive-neighborhood LLMMSE filter. Figure courtesy of M. Ciuc, Laboratorul de Analiza și Prelucrarea Imaginilor, Universitatea Politehnica București, Bucharest, Romania.

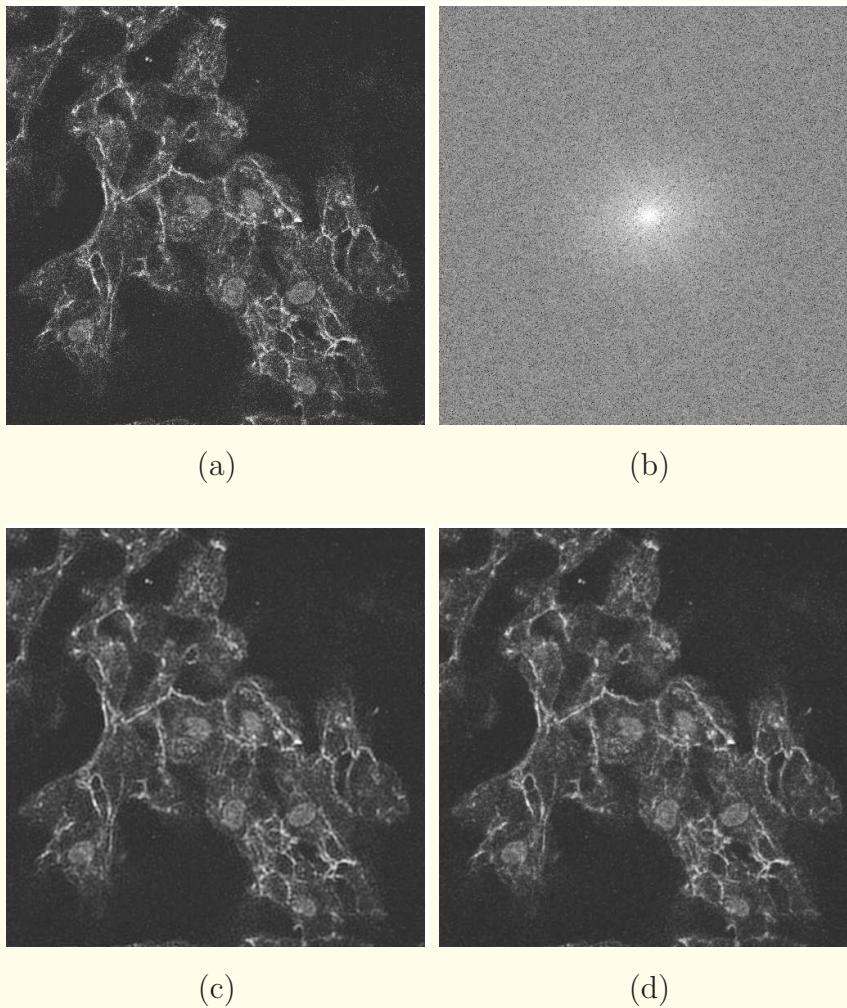


Figure 3.67: (a) The single-frame acquisition of the composite image of the nucleus pulposus of a dog, as in Figure 3.64 (a). (b) Fourier log-magnitude spectrum of the image in (a). Results of filtering the image in (a) with: (c) the ideal lowpass filter with cutoff $D_0 = 0.4$, as in Figure 3.28 (a); and (d) the Butterworth lowpass filter with cutoff $D_0 = 0.4$ and order $n = 2$, as in Figure 3.28 (b).



3.9 Application: Noise Reduction in Nuclear Medicine Imaging

Nuclear medicine images are typically acquired under low-photon conditions, which leads to a significant presence of Poisson noise.

Counting the photons emitted over long periods of time reduces the effect of noise and improves the quality of the image.

However, imaging over long periods of time may not be feasible due to motion artifacts and various practical limitations.

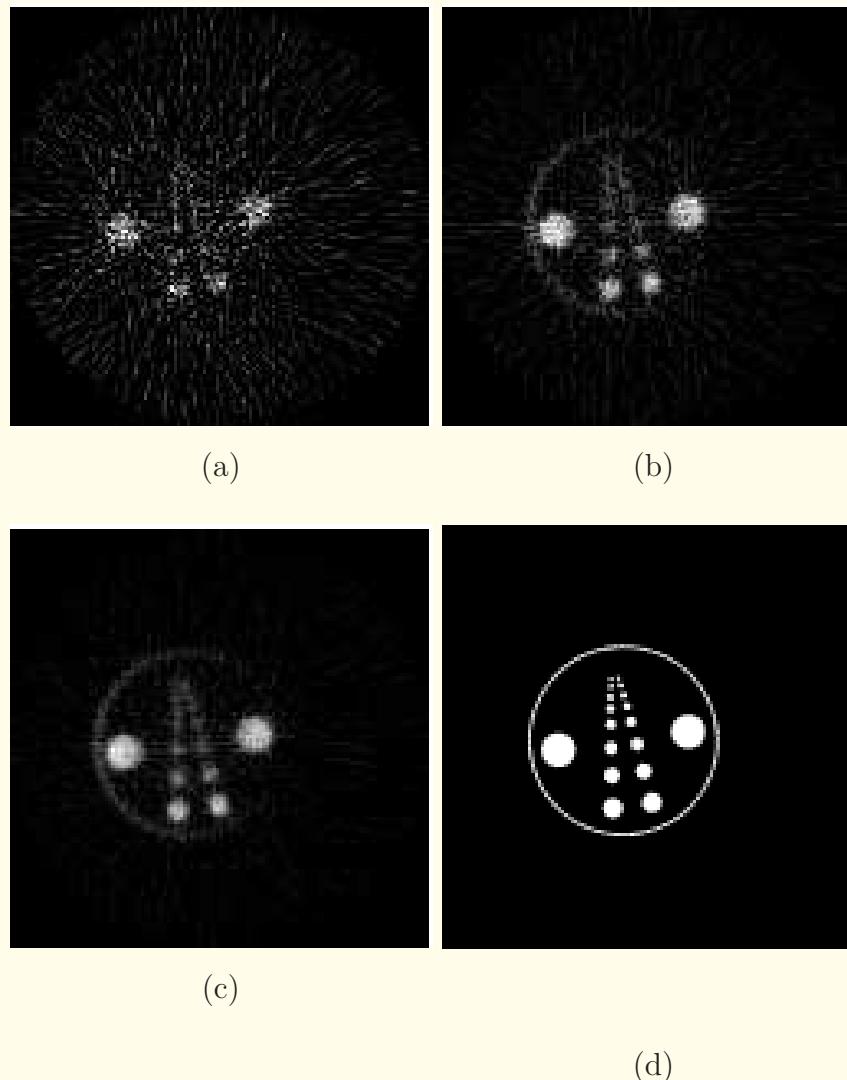


Figure 3.68: 128×128 SPECT images of a resolution phantom obtained by counting photons over: (a) 2 s; (b) 15 s; and (c) 40 s. (d) Schematic representation of the section. Images courtesy of L.J. Hahn, Foothills Hospital, Calgary.

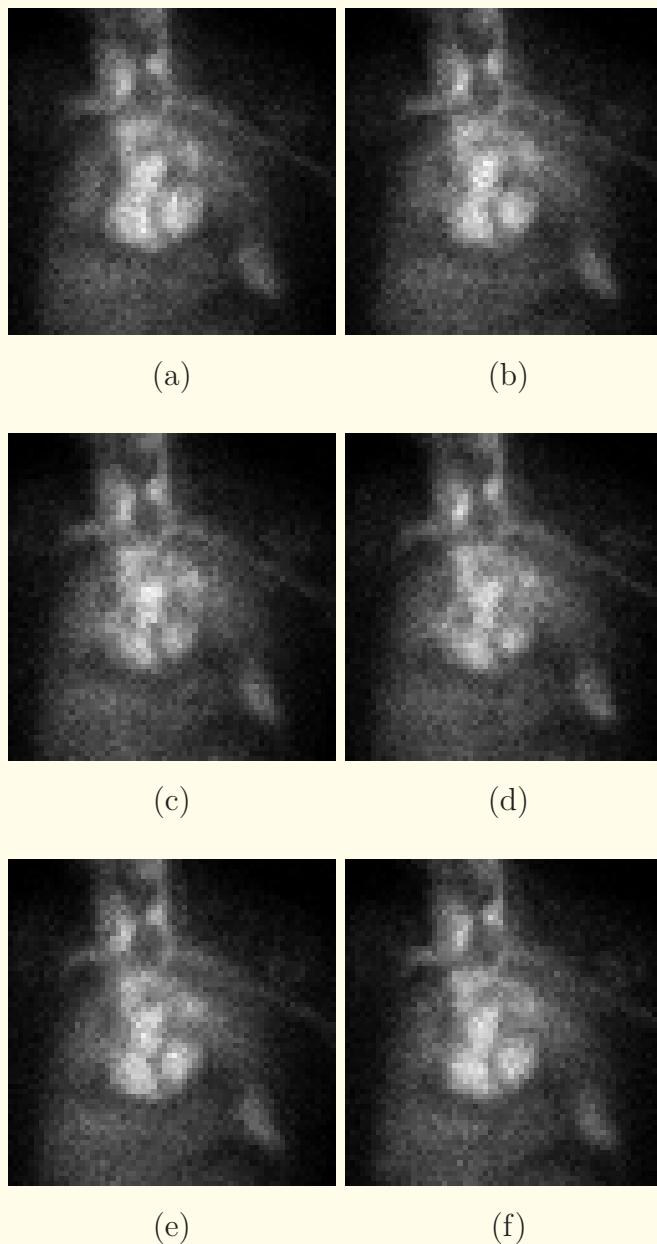


Figure 3.69: (a) 64×64 gated blood-pool images at six phases of the cardiac cycle, obtained by averaging over 16 cardiac cycles. Images courtesy of L.J. Hahn, Foothills Hospital, Calgary.

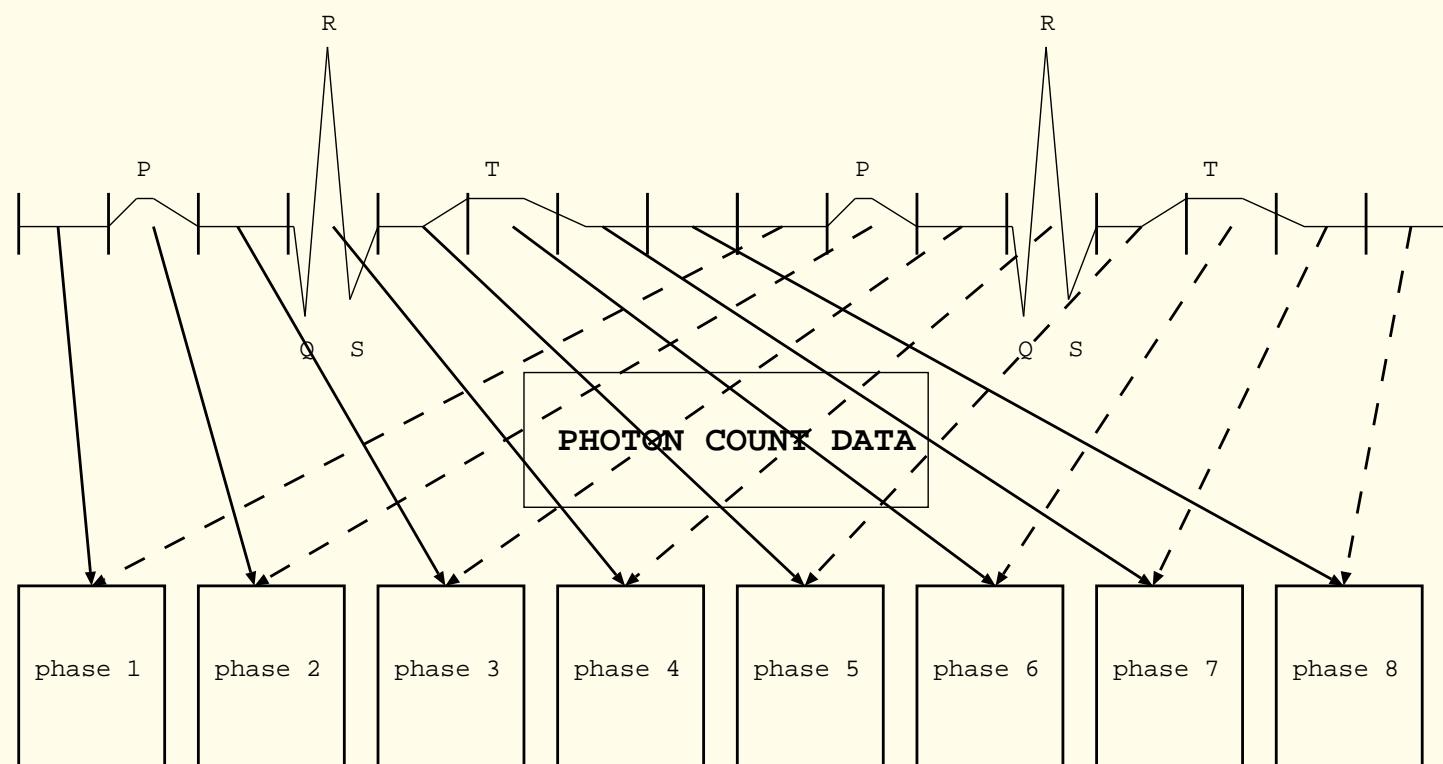


Figure 3.70: Use of the ECG signal in synchronized averaging or accumulation of photon counts in gated blood-pool imaging. Two cycles of cardiac activity are shown by the ECG signal. The ECG waves have the following connotation: P — atrial contraction; QRS — ventricular contraction (systole); T — ventricular relaxation (diastole). Eight frames representing the gated images are shown over each cardiac cycle. Counts over the same phase of the cardiac cycle are added to the same frame over several cardiac cycles.







4

Image Enhancement

Practical factors lead to images with less than the desired levels of contrast, visibility of detail, or overall quality.

In several situations, the understanding of the exact cause of the loss of quality is limited or nonexistent:
attempt to improve or enhance the quality of the image on hand using several techniques applied in an *ad hoc* manner.



In some applications, a nonspecific improvement in the general appearance of the given image may suffice.

A few biomedical imaging situations and applications where enhancement of the feature of interest would be desirable are:

- Microcalcifications in mammograms.
- Lung nodules in chest X-ray images.
- Vascular structure of the brain.
- Hair-line fractures in the ribs.



Some features could be difficult to see in the given image due to their small size, subtlety, small differences in characteristics with respect to their surrounding structures, or low contrast;

others could be rendered not readily visible due to superimposed structures in planar images.

Desired: Enhancement of contrast, edges, and detail visibility
without causing any distortion or artifacts.



4.1 Digital Subtraction Angiography

An X-ray contrast agent (such as an iodine compound) is injected so as to increase the density (attenuation coefficient) of the blood within a certain organ or system of interest.

A number of X-ray images are taken as the contrast agent spreads through the arterial network and before the agent is dispersed via circulation throughout the body.

An image taken before the injection of the agent is used as the “mask” or reference image,

and subtracted from the “live” images obtained with the agent in the system to obtain enhanced images of the arterial system.



Mathematical procedure:

$$\mathbf{f} = \alpha \mathbf{f}_1 - \beta \mathbf{f}_2, \text{ or}$$

$$f(m, n) = \alpha f_1(m, n) - \beta f_2(m, n), \quad (4.1)$$

where \mathbf{f}_1 is the live image, \mathbf{f}_2 is the mask image, α and β are weighting factors (if required), and \mathbf{f} is the result of DSA.

Limitations: Artifacts due to movement.

Allergic reactions to contrast agent.

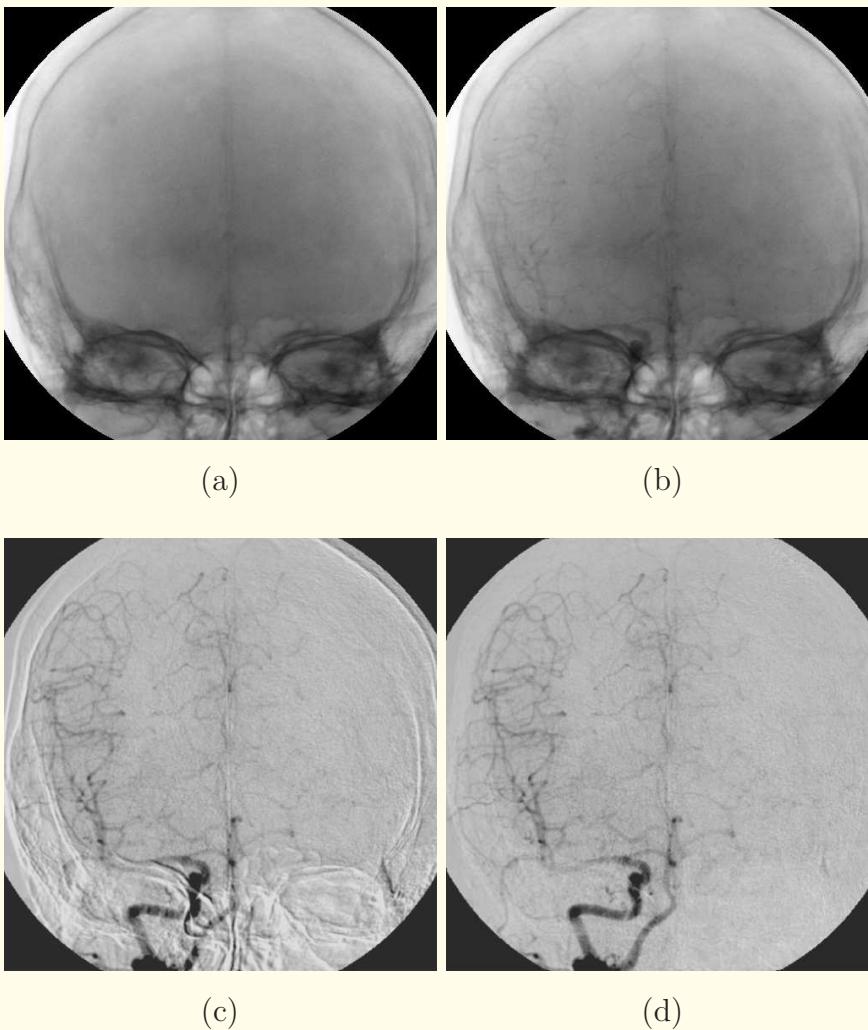


Figure 4.1: (a) Mask image of the head of a patient for DSA. (b) Live image. (c) DSA image of the cerebral artery network. (d) DSA image after correction of motion artifacts. Image data courtesy of E.H.W. Meijering and M.A. Viergever, Image Sciences Institute, University Medical Center Utrecht, Utrecht, The Netherlands. Reproduced with permission from E.H.W. Meijering, K.J. Zuiderveld, and M.A. Viergever, “Image registration for digital subtraction angiography”, *International Journal of Computer Vision*, 31(2/3): 227 – 246, 1999. © Kluwer Academic Publishers.



4.2 Dual-energy and Energy-subtraction X-ray Imaging

Different materials have varying energy-dependent X-ray attenuation coefficients.

X-ray measurements or images obtained at multiple energy levels (also known as energy-selective imaging) could be combined to derive information about the distribution of specific materials.

Weighted combinations of multiple-energy images may be obtained to display soft-tissue and hard-tissue details separately.

Disadvantages: need to subject the patient to two or more X-ray exposures (at different energy or kV).

Motion artifacts.



Energy-subtraction imaging using dual-plate CR:

The plates are separated by a copper filter.

First plate: full-spectrum X-ray image.

Copper filter passes only high-energy X-ray components.

Bones and calcium-containing structures would have preferentially absorbed the low-energy components of the X rays.

High-energy components would have passed through low-density tissues with little attenuation.

Second plate: information related to denser tissues.

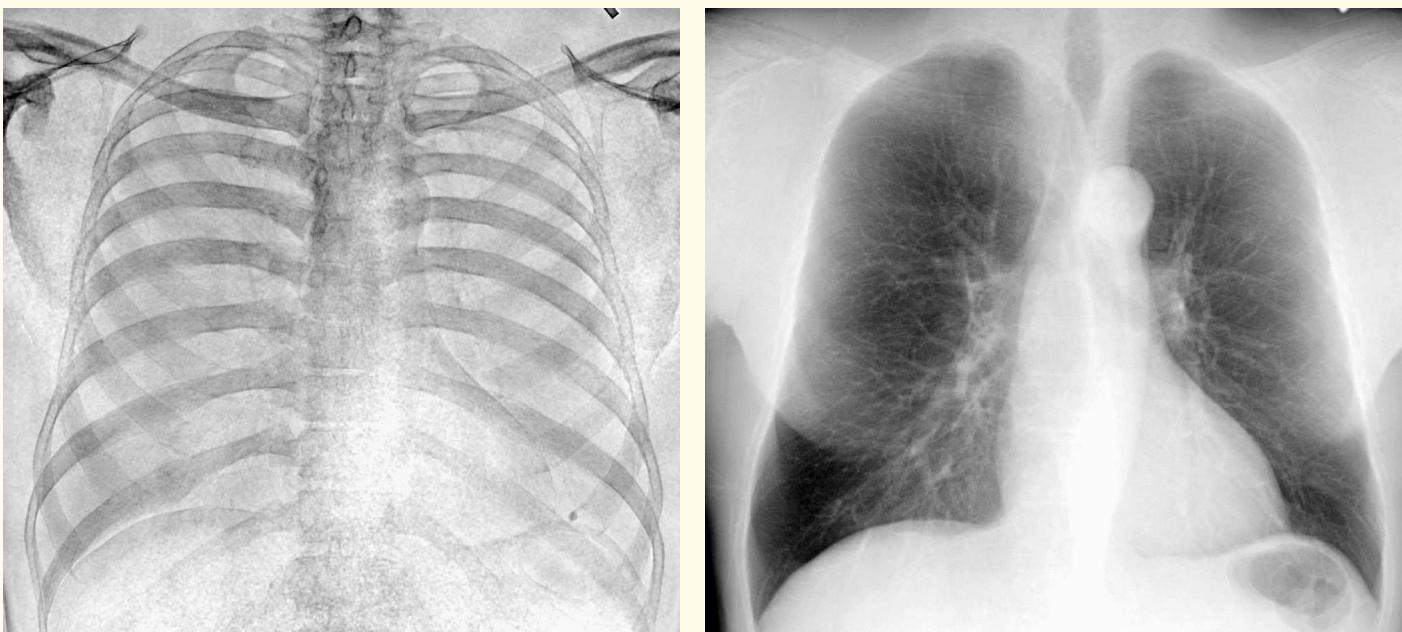


The plates capture two views derived from the same X-ray beam; the patient is not subjected to two different imaging exposures.

Weighted subtraction of the two images provides various results that can demonstrate soft or hard tissues in enhanced detail.



Figure 4.2: Full-spectrum PA chest image (CR) of a patient. See also Figure 4.3. Image courtesy of H. MacMahon, University of Chicago, Chicago, IL. Reproduced with permission from H. MacMahon, “Improvement in detection of pulmonary nodules: Digital image processing and computer-aided diagnosis”, *RadioGraphics*, 20(4): 1169–1171, 2000. © RSNA.



(a)

(b)

Figure 4.3: (a) Bone-detail image, and (b) soft-tissue detail image obtained by energy subtraction. See also Figure 4.2. Images courtesy of H. MacMahon, University of Chicago, Chicago, IL. Reproduced with permission from H. MacMahon, “Improvement in detection of pulmonary nodules: Digital image processing and computer-aided diagnosis”, *RadioGraphics*, 20(4): 1169–1171, 2000. © RSNA.



4.3 Temporal Subtraction

Time-lapse subtraction of images useful in detecting normal or pathological changes that have occurred over a period of time.

Normal anatomic structures are suppressed and pathological changes are enhanced.

Registration of the images is crucial in temporal subtraction; misregistration could lead to artifacts.

Geometric transformation techniques are useful in matching landmark features that are not expected to have changed in the interval between the two imaging sessions.



4.4 Gray-scale Transforms

4.4.1 *Gray-scale thresholding*

When the gray levels of the objects of interest in an image are known, the image may be thresholded to obtain a variety of images that can display selected features of interest.

$$g(m, n) = \begin{cases} 0 & \text{if } f(m, n) \leq L_1 \\ 255 & \text{if } f(m, n) > L_1 \end{cases}. \quad (4.2)$$

The result is a bilevel or binary image.



If pixels less than L_1 were to be considered as noise
(or features of no interest),

and gray levels of objects of interest are greater than L_1 :

$$g(m, n) = \begin{cases} 0 & \text{if } f(m, n) \leq L_1 \\ f(m, n) & \text{if } f(m, n) > L_1 \end{cases}. \quad (4.3)$$

Displays features of interest including their gray-level variations.

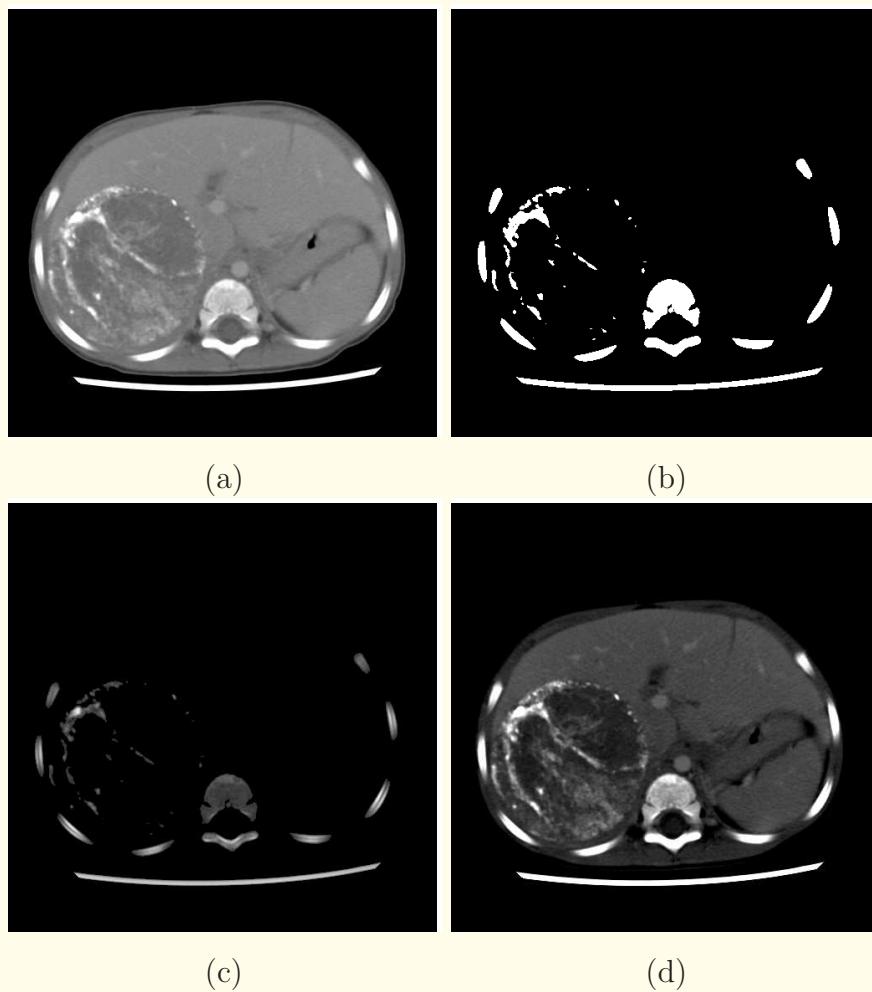


Figure 4.4: (a) CT image of a patient with neuroblastoma. The tumor, which appears as a large circular region on the left-hand side of the image, includes calcified tissues that appear as bright regions. The HU range of $[-200, 400]$ has been linearly mapped to the display range of $[0, 255]$. (b) The image in (a) thresholded at the level of 200 HU as in Equation 4.2. Values above 200 HU appear as white, and values below this threshold appear as black. (c) The image in (a) thresholded at the level of 200 HU as in Equation 4.3. Values above 200 HU appear at their original level, and values below this threshold appear as black. (d) The HU range of $[0, 400]$ has been linearly mapped to the display range of $[0, 255]$ as in Equation 4.4. Pixels corresponding to tissues lighter than water appear as black. Pixels greater than 400 HU are saturated at the maximum gray level of 255 .



4.4.2 Gray-scale windowing

If a given image $f(m, n)$ has all of its pixel values in a narrow range of gray levels, or

if certain details of particular interest within the image occupy a narrow range of gray levels,

stretch the range of interest to the full range of display available:

$$g(m, n) = \begin{cases} 0 & \text{if } f(m, n) \leq f_1 \\ \frac{f(m, n) - f_1}{f_2 - f_1} & \text{if } f_1 < f(m, n) < f_2 \\ 1 & \text{if } f(m, n) \geq f_2 \end{cases} . \quad (4.4)$$



$g(m, n)$ is the windowed image to be displayed, with its gray-scale normalized to the range $[0, 1]$;

$[f_1, f_2]$ is the range of the original gray-level values to be displayed in the output after stretching to the full range.

Pixels below f_1 will be eliminated (rendered black);

those above the upper limit f_2 will be saturated (rendered white).

The details within the range $[f_1, f_2]$ will be displayed with increased contrast and latitude.

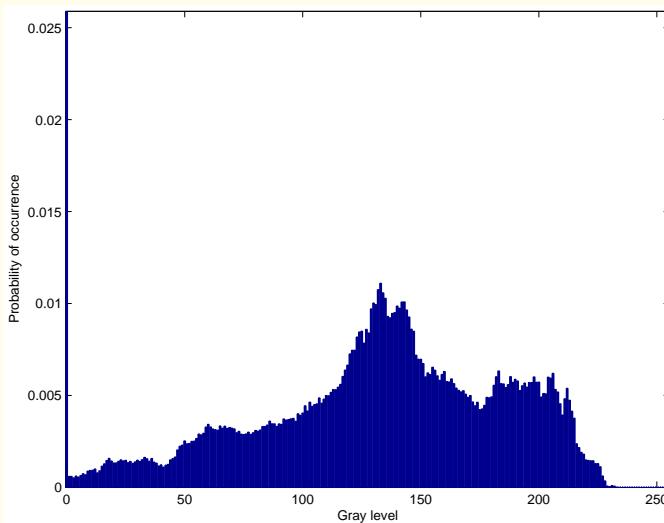


(a)

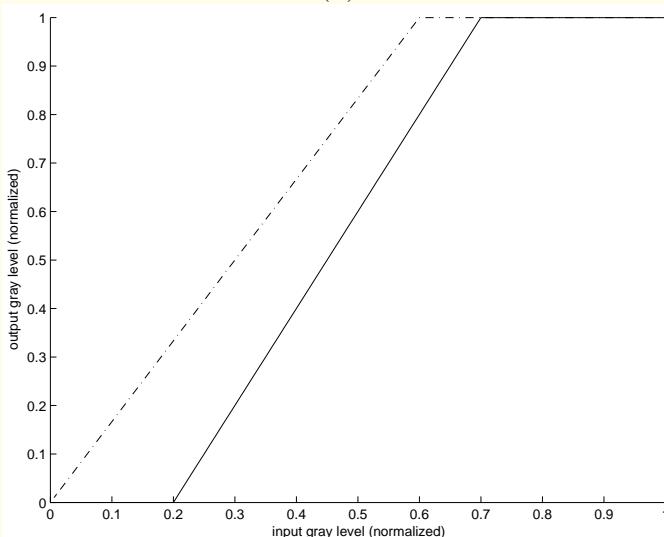
(b)

(c)

Figure 4.5: (a) Part of a chest X-ray image. The histogram of the image is shown in Figure 4.6 (a). (b) Image in (a) enhanced by linear mapping of the range $[0, 0.6]$ to $[0, 1]$. (c) Image in (a) enhanced by linear mapping of the range $[0.2, 0.7]$ to $[0, 1]$. See Figure 4.6 (b) for plots of the LUTs.

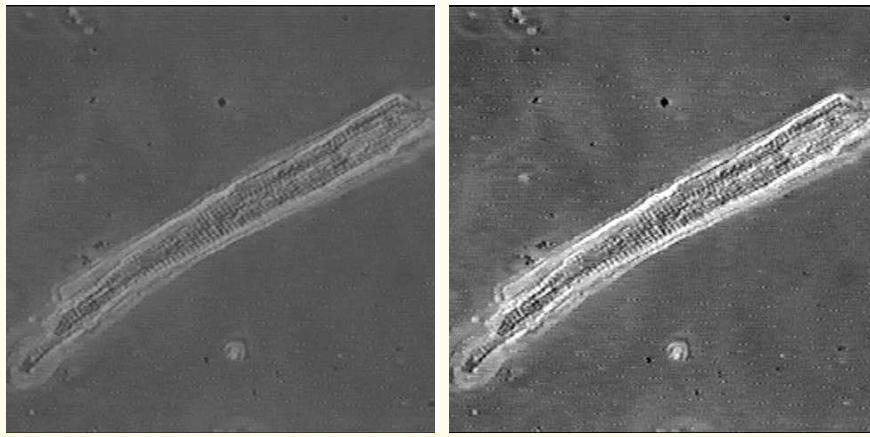


(a)



(b)

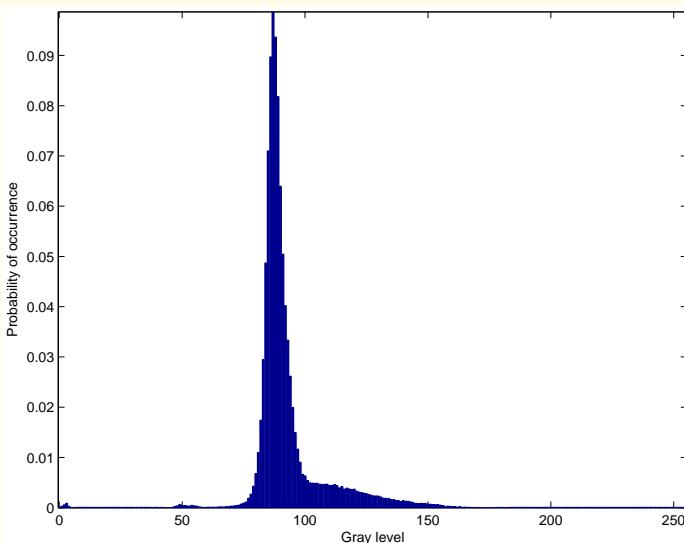
Figure 4.6: (a) Normalized histogram of the chest X-ray image in Figure 4.5 (a); entropy = 7.55 bits. (b) Linear density-windowing transformations that map the ranges $[0, 0.6]$ to $[0, 1]$ (dash-dot line) and $[0.2, 0.7]$ to $[0, 1]$ (solid line).



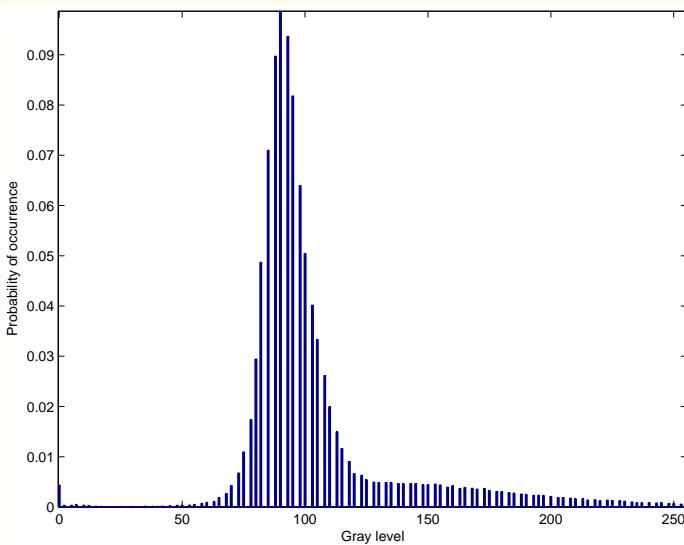
(a)

(b)

Figure 4.7: (a) Image of a myocyte as acquired originally. (b) Image in (a) enhanced by linear mapping of the normalized range $[0.2, 0.6]$ to $[0, 1]$. See Figure 4.8 for the histograms of the images.



(a)

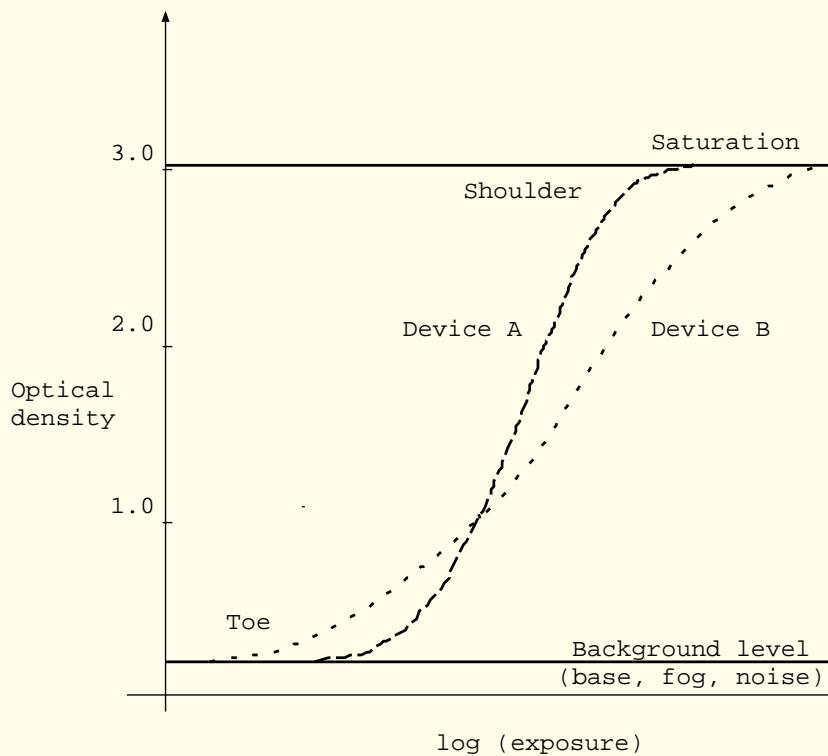


(b)

Figure 4.8: Normalized histograms of (a) the image in Figure 4.7 (a), entropy = *4.96 bits*; and (b) the image in Figure 4.7 (b), entropy = *4.49 bits*.



4.4.3 Gamma correction



The slope of the H-D curve is known as gamma or γ .



System with large γ : image with high contrast;

may not utilize the full range of the available gray scale.

Small γ : image with wide latitude but poor contrast.

Gamma correction is a nonlinear transformation process to change the contrast and latitude of gray scale in the image.

$$g(m, n) = [f(m, n)]^\gamma, \quad (4.5)$$

$f(m, n)$: gray scale normalized to the range $[0, 1]$.



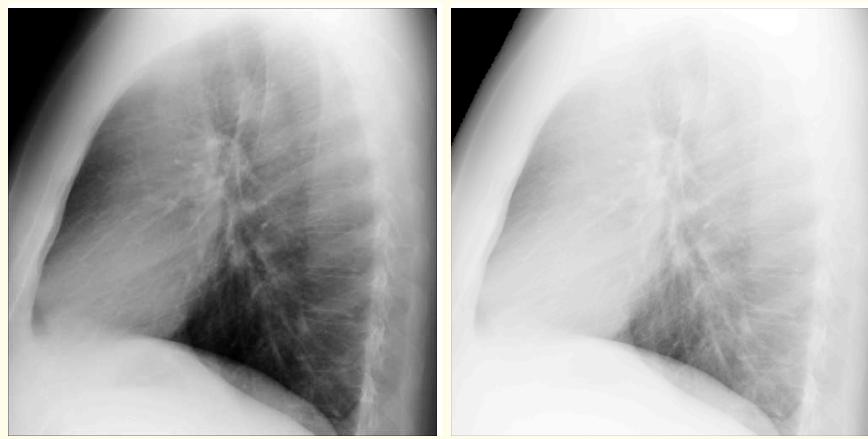
Another definition:

$$g(m, n) = \exp \left[\frac{\ln\{f(m, n)\}}{\gamma} \right], \quad (4.6)$$

equivalent to Equation 4.5 if gray levels not normalized,

that is, the gray levels were to remain in a range such as $0 - 255$.

Gray-scale windowing could also be incorporated.



(a)

(b)



(c)

Figure 4.9: (a) Part of a chest X-ray image. (b) Image in (a) enhanced with $\gamma = 0.3$. (c) Image in (a) enhanced with $\gamma = 2.0$. See Figure 4.10 for plots of the gamma-correction transforms (LUTs).

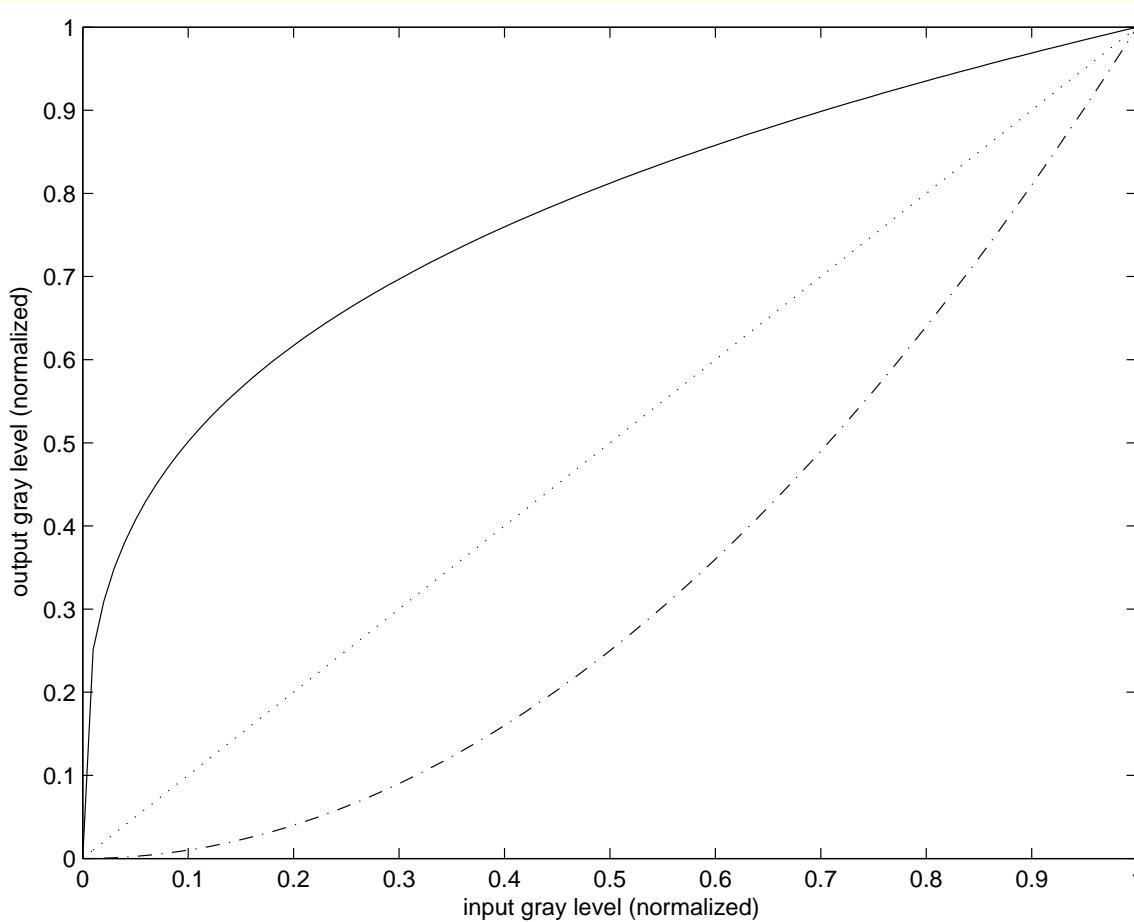


Figure 4.10: Gamma-correction transforms with $\gamma = 0.3$ (solid line), $\gamma = 1.0$ (dotted line), and $\gamma = 2.0$ (dash-dot line).



4.5 Histogram Transformation

Image histogram normalized and interpreted as a PDF.

Based upon certain principles of information theory, maximal information is conveyed when the PDF of a process is uniform:

the corresponding image has all possible gray levels with equal probability of occurrence.

The entropy of the PDF is at its maximum.

The technique of histogram equalization has been proposed as a method to enhance the appearance of an image.



4.5.1 Histogram equalization

Consider an image $f(m, n)$ of size $M \times N$ pixels, with gray levels $l = 0, 1, 2, \dots, L - 1$.

Histogram of the image = $P_f(l)$.

Normalize the gray levels by dividing by the maximum level available or permitted, as $r = \frac{l}{L-1}$, such that $0 \leq r \leq 1$.

Let $p_f(r)$ be the normalized histogram or PDF.



If we were to apply a transformation $s = T(r)$ to the random variable r , the PDF of the new variable s is given by

$$p_g(s) = p_f(r) \frac{dr}{ds} \Big|_{r=T^{-1}(s)} . \quad (4.7)$$

Resulting image $g(m, n)$ with normalized gray levels
 $0 \leq s \leq 1$.



Consider the transformation

$$s = T(r) = \int_0^r p_f(w) dw; \quad 0 \leq r \leq 1. \quad (4.8)$$

This is the cumulative (probability) distribution function of r .

$T(r)$ has the following important and desired properties:

- $T(r)$ is single-valued and monotonically increasing over the interval $0 \leq r \leq 1$.

Necessary to maintain the black-to-white transition order between the original and processed images.

- $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$.

Required in order to maintain the same range of values in the input and output images.



It follows that $\frac{ds}{dr} = p_f(r)$.

$$p_g(s) = \left[p_f(r) \frac{1}{p_f(r)} \right]_{r=T^{-1}(s)} = 1; \quad 0 \leq s \leq 1. \quad (4.9)$$

Thus, $T(r)$ *equalizes* the histogram of the given image;

the histogram or PDF of the resulting image $g(m, n)$ is uniform.

A uniform PDF has maximal entropy.



Discrete version of histogram equalization:

Digital image $f(m, n)$ with $P = MN$ pixels and L gray levels $r_k, k = 0, 1, \dots, L - 1, 0 \leq r_k \leq 1$, occurring n_k times.

The PDF may be approximated by the histogram

$$p_f(r_k) = \frac{n_k}{P}; \quad k = 0, 1, \dots, L - 1. \quad (4.10)$$

The histogram-equalizing transformation is approximated by

$$s_k = T(r_k) = \sum_{i=0}^k p_f(r_i) = \sum_{i=0}^k \frac{n_i}{P}; \quad k = 0, 1, \dots, L - 1. \quad (4.11)$$



Discrete histogram transformation may yield values of s_k that may not equal the available quantized gray levels.

The values will have to be quantized, and hence the output image may only have an approximately uniform histogram.

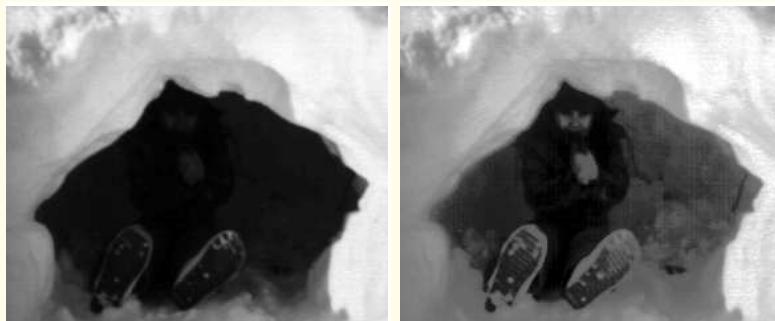
The resulting values in the range $[0, 1]$ have to be scaled to the display range, such as $[0, 255]$.



Histogram equalization is usually implemented via an LUT that lists the related (s_k, r_k) pairs as given by Equation 4.11.

A quantized histogram-equalizing transformation may contain several many-to-one gray-level mappings:

this renders the transformation nonunique and irreversible.



(a)



(b)

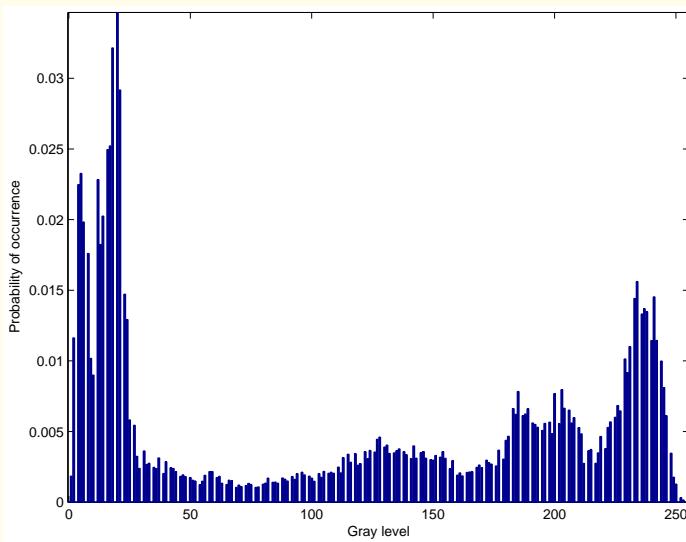


(c)

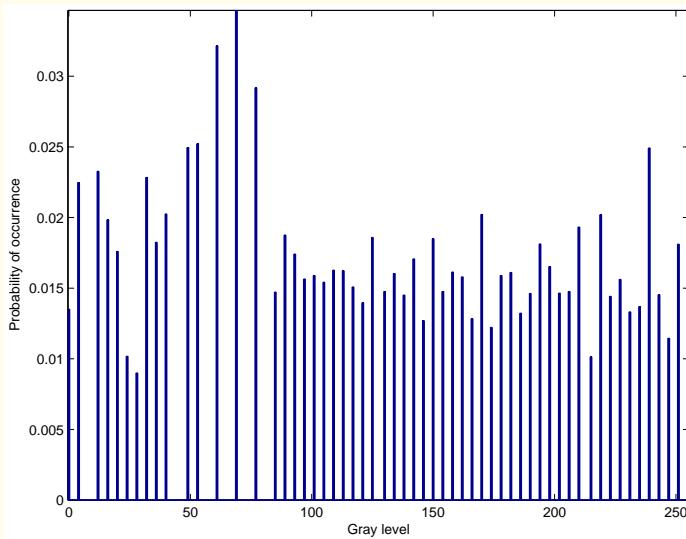


(d)

Figure 4.11: (a) Image of a girl in a snow cave (240×288 pixels). (b) Result of histogram equalization. (c) Result of linear mapping (windowing) of the range $[0, 23]$ to $[0, 255]$. (d) Result of gamma correction with $\gamma = 0.3$. Image courtesy of W.M. Morrow.



(a)



(b)

Figure 4.12: Normalized histograms of (a) the image in Figure 4.11 (a), entropy = *6.93 bits*; and (b) the image in Figure 4.11 (b), entropy = *5.8 bits*. See also Figure 4.13.

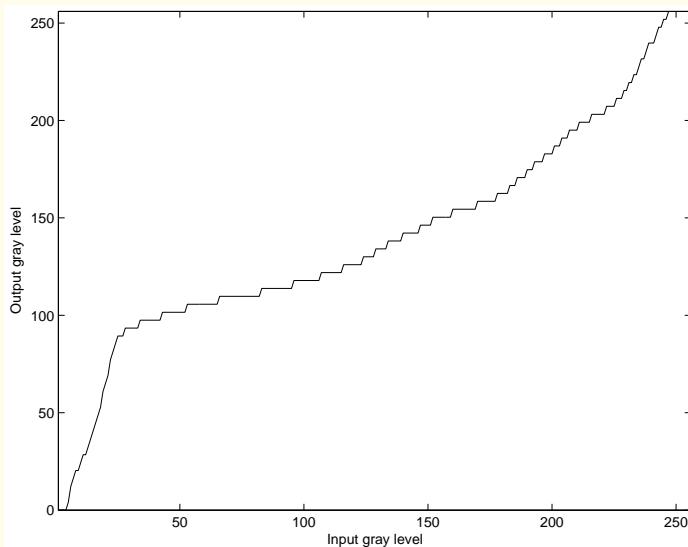
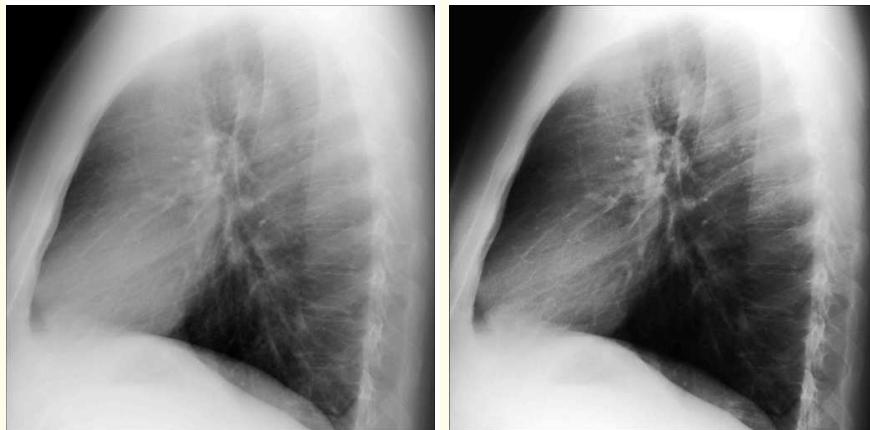


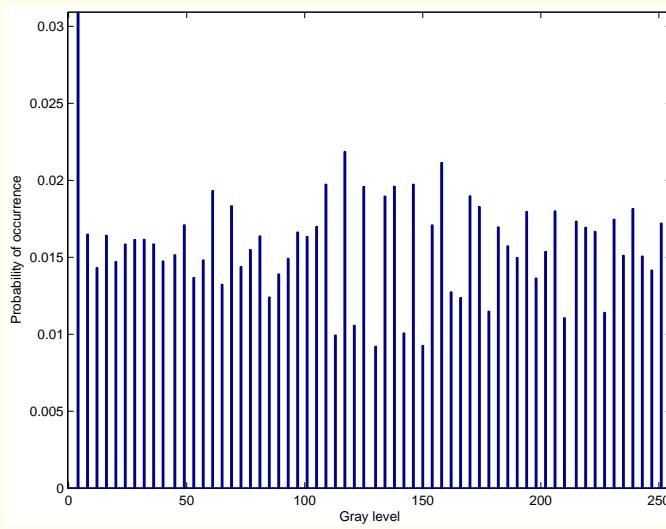
Figure 4.13: Histogram-equalizing transform (LUT) for the image in Figure 4.11 (a); see Figure 4.12 for the histograms of the original and equalized images.



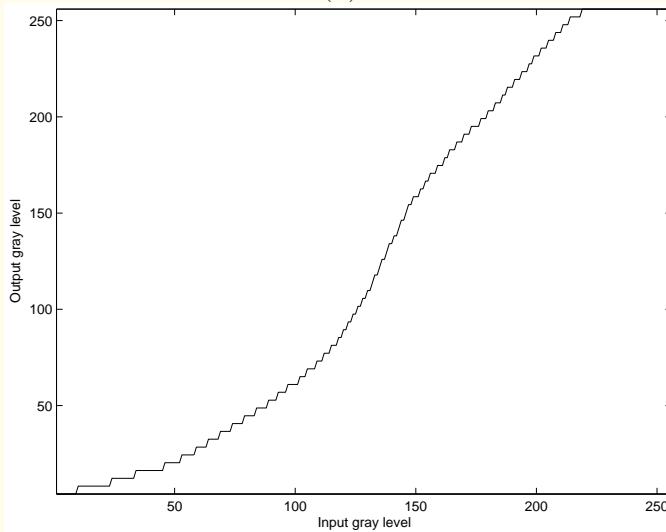
(a)

(b)

Figure 4.14: (a) Part of a chest X-ray image. The histogram of the image is shown in Figure 4.6 (a). (b) Image in (a) enhanced by histogram equalization. The histogram of the image is shown in Figure 4.15 (a). See Figure 4.15 (b) for a plot of the LUT.



(a)



(b)

Figure 4.15: (a) Normalized histogram of the histogram-equalized chest X-ray image in Figure 4.14 (b); entropy = *5.95 bits*. (b) The histogram-equalizing transformation (LUT). See Figure 4.6 (a) for the histogram of the original image.

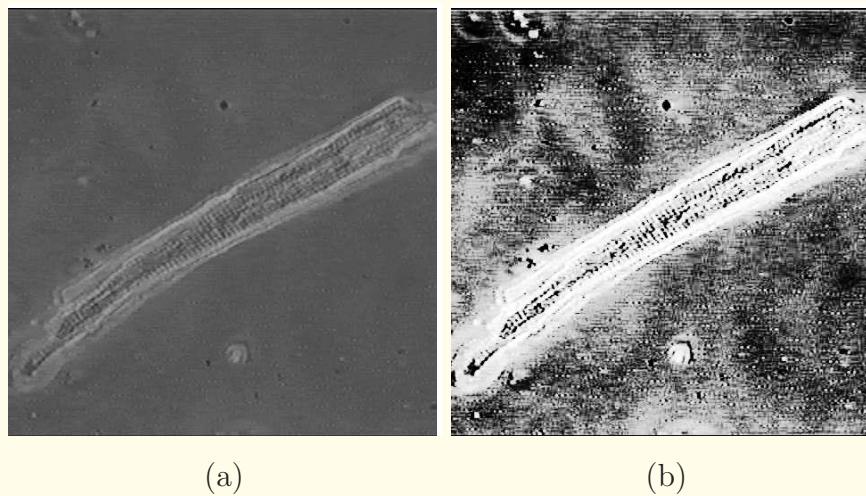


Figure 4.16: (a) Image of a myocyte. The histogram of the image is shown in Figure 4.8 (a). (b) Image in (a) enhanced by histogram equalization. The histogram of the image is shown in Figure 4.17 (a). See Figure 4.17 (b) for a plot of the LUT.

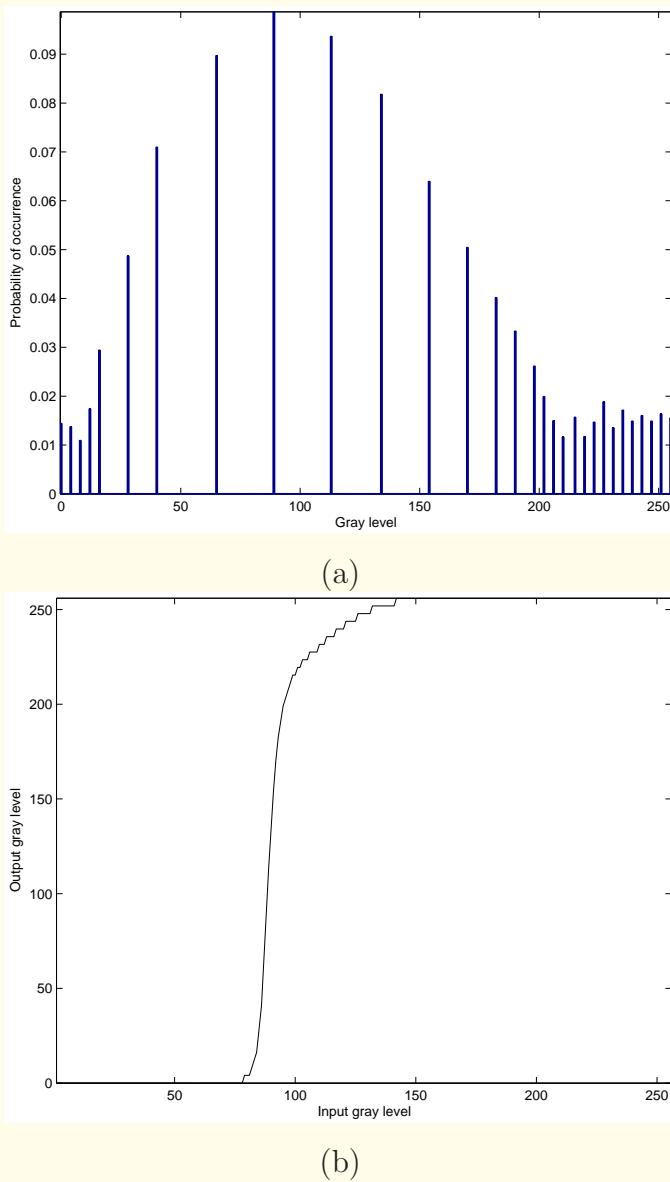


Figure 4.17: (a) Normalized histogram of the histogram-equalized myocyte image in Figure 4.16 (b). (b) The histogram-equalizing transformation (LUT). See Figure 4.8 (a) for the histogram of the original image.



4.5.2 Histogram specification

Major limitations of histogram equalization:

can provide only one output image,

user has no control over the procedure or the result.

Histogram specification: a series of histogram-equalization steps used to obtain an image with a histogram that is expected to be close to a *prespecified* histogram.

By specifying several histograms, it is possible to obtain a range of several enhanced images.



Desired or specified normalized histogram is $p_d(t)$,

with the desired image d having the normalized gray levels
 $t = 0, 1, 2, \dots, L - 1$.

Given image f with PDF $p_f(r)$ histogram-equalized by the transformation

$$s = T_1(r) = \int_0^r p_f(w) dw; \quad 0 \leq r \leq 1, \quad (4.12)$$

to obtain the image g with the normalized gray levels s .



We may also derive a histogram-equalizing transform for the desired (but as yet unavailable) image as

$$q = T_2(t) = \int_0^t p_d(w) dw; \quad 0 \leq t \leq 1. \quad (4.13)$$

Note: in order to derive a histogram-equalizing transform, we need only the PDF of the image; the image itself is not needed.

Let the (hypothetical) image so obtained be e , with gray levels q .

$t = T_2^{-1}(q)$ will map the gray levels q back to t .



$p_g(s)$ and $p_e(q)$ are both uniform PDFs, and hence identical.

The desired PDF may, therefore, be obtained by applying the transform T_2^{-1} to s :

$$t = T_2^{-1}(s).$$

Assumed that $T_2^{-1}(s)$ exists, and is a single-valued (unique) transform.



Procedure for histogram specification:

1. Specify desired histogram and derive equivalent PDF $p_d(t)$.
2. Derive the histogram-equalizing transform $q = T_2(t)$.
3. Derive the histogram-equalizing transform $s = T_1(r)$ from the PDF $p_f(r)$ of the given image f .
4. Apply the inverse of the transform T_2 to the PDF obtained in the previous step and obtain $t = T_2^{-1}(s)$.
This step may be directly implemented as $t = T_2^{-1}[T_1(r)]$.
5. Apply the transform as above to the given image f ;
result provides desired image d with specified PDF $p_d(t)$.



4.5.3 *Limitations of global operations*

A global approach to image enhancement ignores the *nonstationary nature of images*.

It is desirable to design *local and adaptive transforms* for effective image enhancement.



4.5.4 Local-area histogram equalization

Ketchum: the histogram of the pixels within a 2D sliding rectangular window, centered at the current pixel being processed, is equalized;

the resulting transform is applied only to the central pixel;

the process is repeated for every pixel in the image.

The window provides the *local context* for the pixel being processed.



Pizer et al.: the histogram-equalizing transforms are computed not for every pixel, but only for a number of nonoverlapping rectangular blocks spanning the image.

The pixels at the center of each block are processed using the corresponding transform.

Pixels that are not at the centers of the blocks are processed using interpolated versions of the transforms corresponding to the four neighboring center pixels.

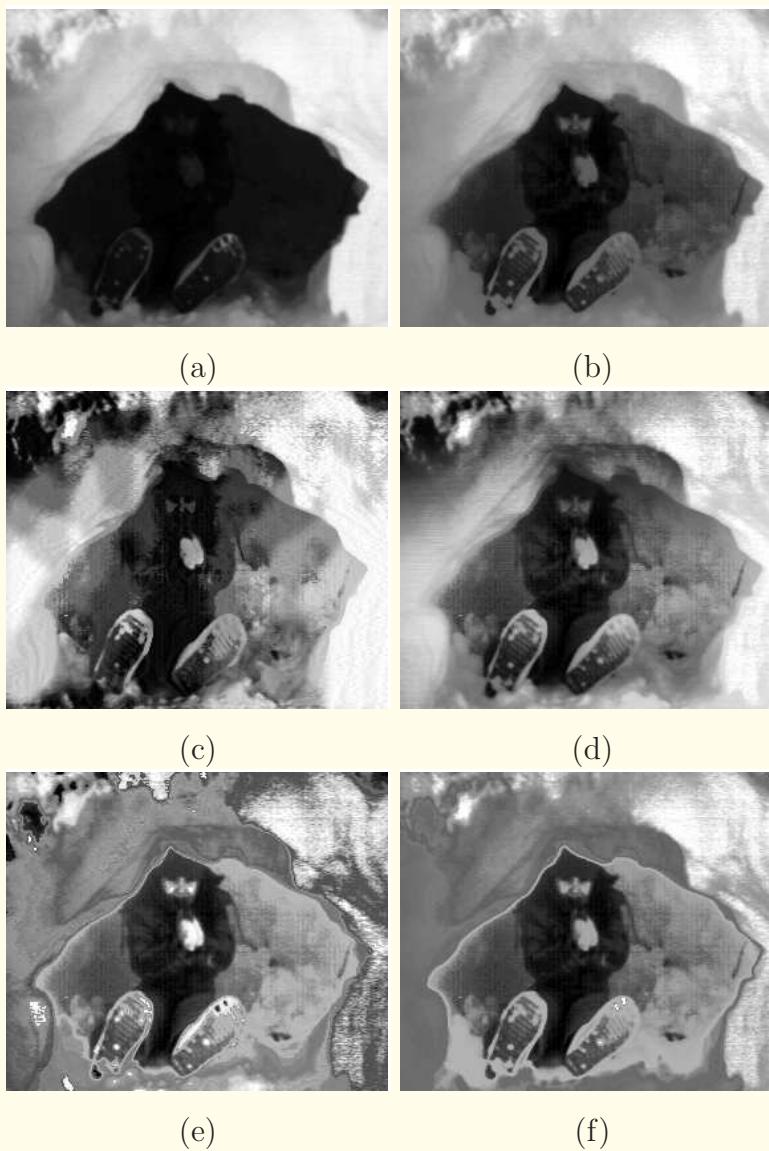


Figure 4.18: (a) Image of a girl in a snow cave (240×288 pixels). (b) Result of global histogram equalization. Results of LAHE with (c) a 11×11 window and (d) a 101×101 window. Results of adaptive-neighborhood histogram equalization with (e) growth tolerance 16 and background width 5 pixels, and (f) growth tolerance 64 and background width 8 pixels. Reproduced with permission from R.B. Paranjape, W.M. Morrow, and R.M. Rangayyan, “Adaptive-neighborhood histogram equalization for image enhancement”, *CVGIP: Graphical Models and Image Processing*, 54(3):259–267, 1992. © Academic Press.



4.5.5 *Adaptive-neighborhood histogram equalization*

A limitation of LAHE lies in the use of rectangular windows:

there is no apparent justification to the choice of the rectangular shape for the moving window.

Success of the method depends significantly upon proper choice of the size of the window;

the use of a fixed window of a prespecified size over an entire image has no particular reasoning.



ANHE: histogram of adaptive-neighborhood foreground region and a background composed of a molded ribbon of pixels.

The method adapts to local details present in the given image; regions of different size and shape are grown for each pixel.

Result of equalizing transform applied only to the seed pixel from where the process was started, and the redundant seed pixels.



(a)



(b)



(c)



(d)

Figure 4.19: (a) A test image and its enhanced versions by: (b) global or full-frame histogram equalization, (c) LAHE, and (d) adaptive-neighborhood histogram equalization. Image courtesy of R.B. Paranjape.



4.6 Convolution Mask Operators

Filtering images using 3×3 convolution masks is popular.

Several masks have been proposed and are in practical use for image enhancement.



4.6.1 *Unsharp masking*

When an image is blurred by some unknown phenomenon:

we could assume that each pixel in the original image contributes, in an additive manner, a certain fraction of its value to the neighboring pixels.

Then, each pixel is composed of its own true value, plus fractional components of its neighbors.

The spreading of the value of a pixel into its neighborhood may be viewed as a local fog or blurred background.



Photographic technique of unsharp masking:

the given degraded image, in its negative form, is first blurred, and a positive transparency is created from the result.

The original negative and the positive are held together, and a (positive) print is made of the combination.

The procedure leads to the subtraction of the local blur or fog component, and hence to an improved and sharper image.



A popular 3×3 convolution mask that mimics unsharp masking:

$$\begin{bmatrix} -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \\ -\frac{1}{8} & 2 & -\frac{1}{8} \\ -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \end{bmatrix}. \quad (4.14)$$

Observe that the net sum of the values in the mask equals unity; therefore, there is no net change in the local average intensity.



Unsharp masking may be generalized to permit the use of other local window sizes and shapes as

$$f_e(m, n) = [g(m, n) - \mu_g(m, n)] + \alpha g(m, n). \quad (4.15)$$

The enhanced image $f_e(m, n)$ is given as a weighted combination of the corresponding pixel $g(m, n)$ in the given degraded image, and the difference between the pixel and the local mean $\mu_g(m, n)$.

The expression is equivalent to the mask in Equation 4.14, with $\alpha = 1$ and the local mean being computed as the average of the eight neighbors of the pixel being processed.



Difference in Equation 4.15: a measure of the local gradient;

because gradients are associated with edges, combining the given image with its local gradient could lead to edge enhancement or high-frequency emphasis.



The unsharp masking filter performs *edge enhancement*.

Strong edges will have a clearly perceptible overshoot and undershoot: a form of ringing artifact.

The unsharp masking operation could lead to negative pixel values in the enhanced image;

user has to decide how to handle this when displaying result.

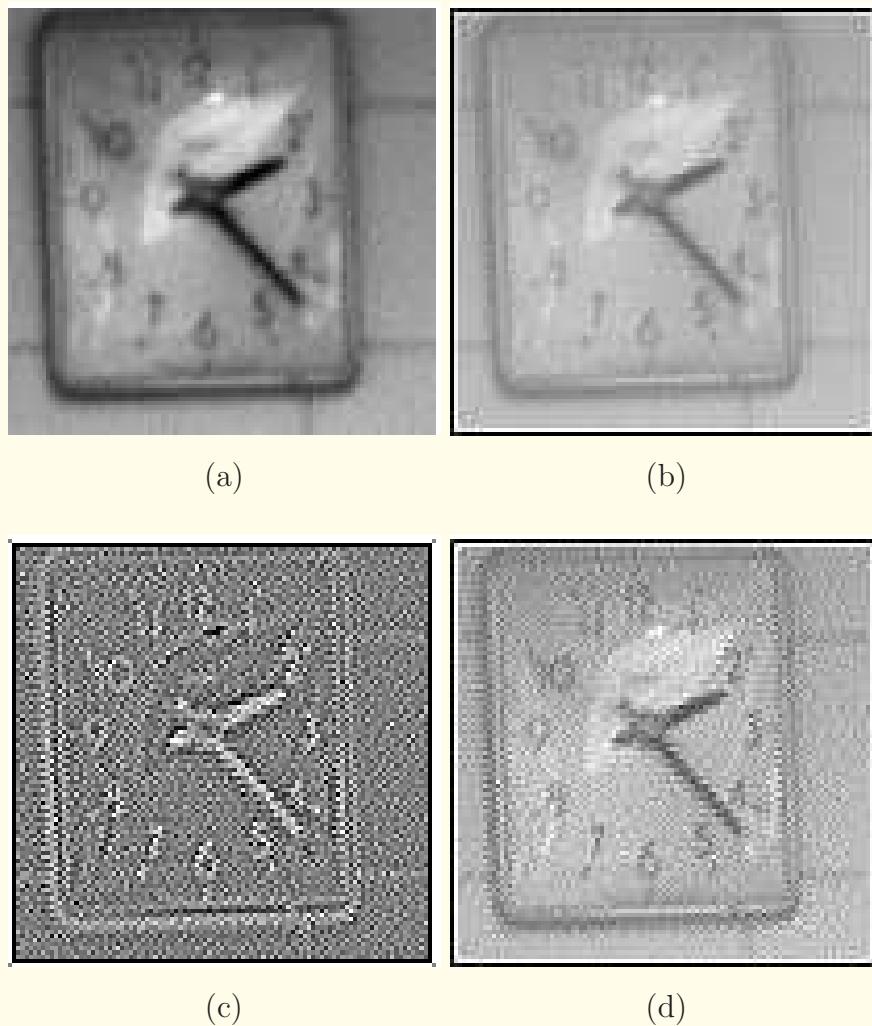


Figure 4.20: (a) Clock test image. (b) Result of unsharp masking; display range $[-50, 250]$ out of $[-68, 287]$. (c) Laplacian (gradient) of the image; display range $[-50, 50]$ out of $[-354, 184]$. (d) Result of the subtracting Laplacian; display range $[-50, 250]$ out of $[-184, 250]$.

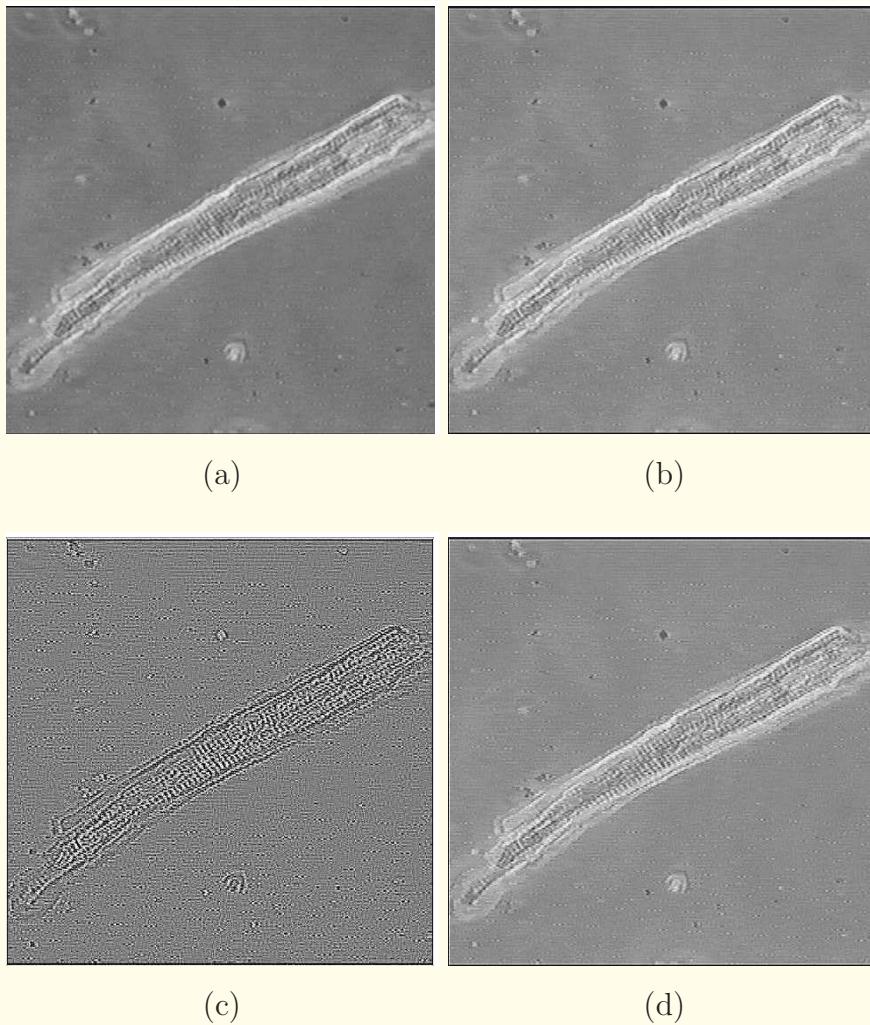


Figure 4.21: (a) Image of a myocyte; the range from the minimum to the maximum of the image has been linearly mapped to the display range [0, 255]. (b) Result of unsharp masking; display range $[-20, 180]$ out of $[-47, 201]$. (c) Laplacian (gradient) of the image; display range $[-20, 20]$ out of $[-152, 130]$. (d) Result of the subtracting Laplacian; display range $[-50, 200]$ out of $[-130, 282]$.

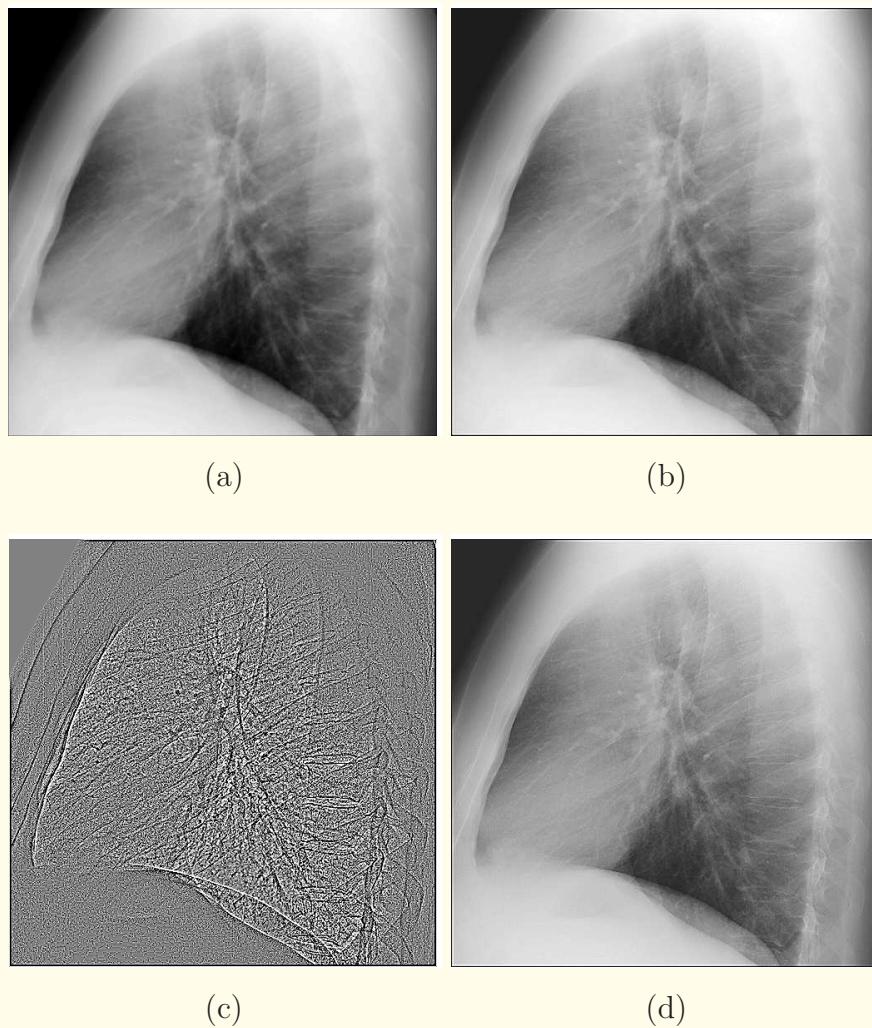


Figure 4.22: (a) Part of a chest X-ray image. (b) Result of unsharp masking; display range $[-30, 230]$ out of $[-59, 264]$. (c) Laplacian (gradient) of the image; display range $[-5, 5]$ out of $[-134, 156]$. (d) Result of the subtracting Laplacian; display range $[-50, 250]$ out of $[-156, 328]$.

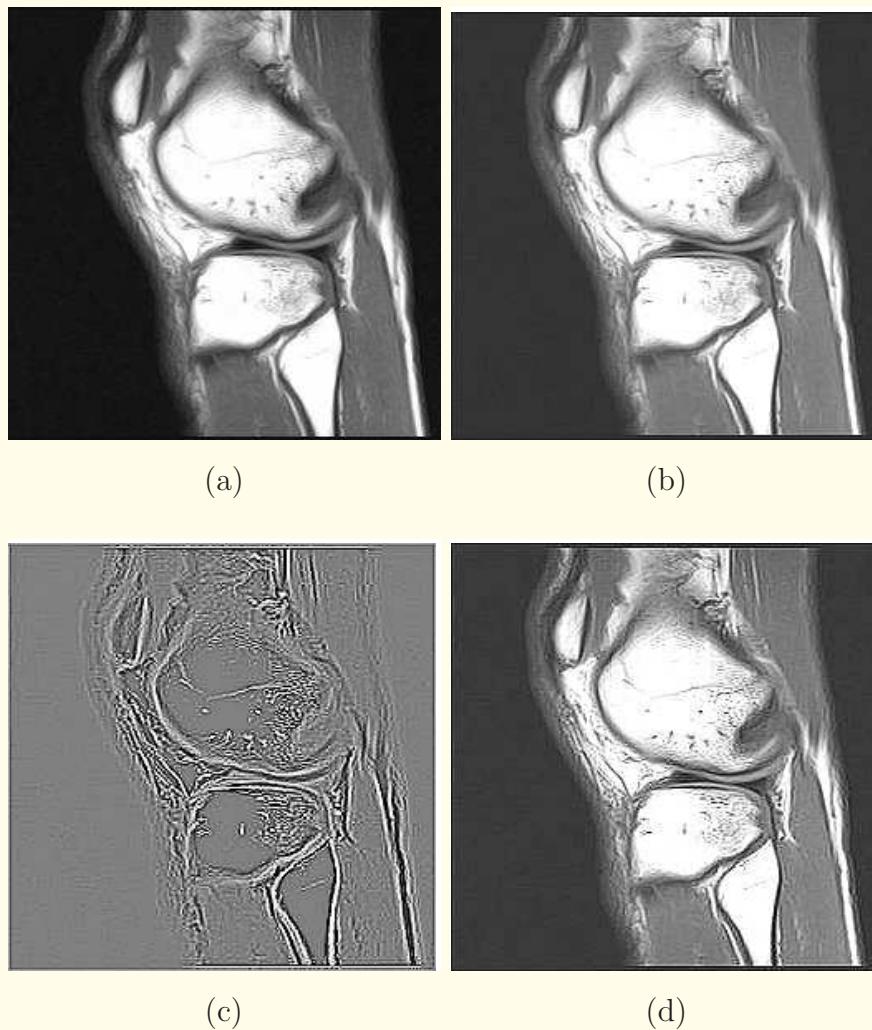


Figure 4.23: (a) MR image of a knee. (b) Result of unsharp masking; display range $[-40, 250]$ out of $[-72, 353]$. (c) Laplacian (gradient) of the image; display range $[-50, 50]$ out of $[-302, 365]$. (d) Result of the subtracting Laplacian; display range $[-50, 250]$ out of $[-261, 549]$.

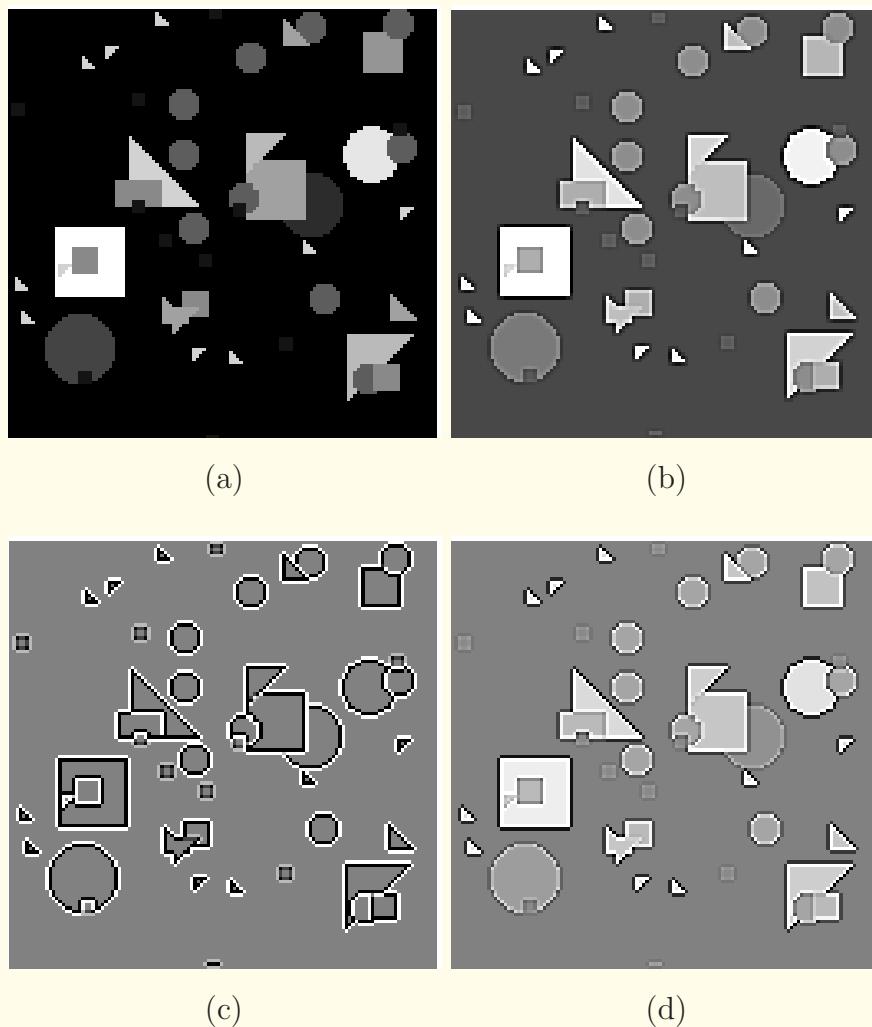


Figure 4.24: (a) Shapes test image. (b) Result of unsharp masking; display range $[-100, 250]$ out of $[-130, 414]$. See also Figure 4.25. (c) Laplacian (gradient) of the image; display range $[-50, 50]$ out of $[-624, 532]$. (d) Result of the subtracting Laplacian; display range $[-300, 300]$ out of $[-532, 832]$.

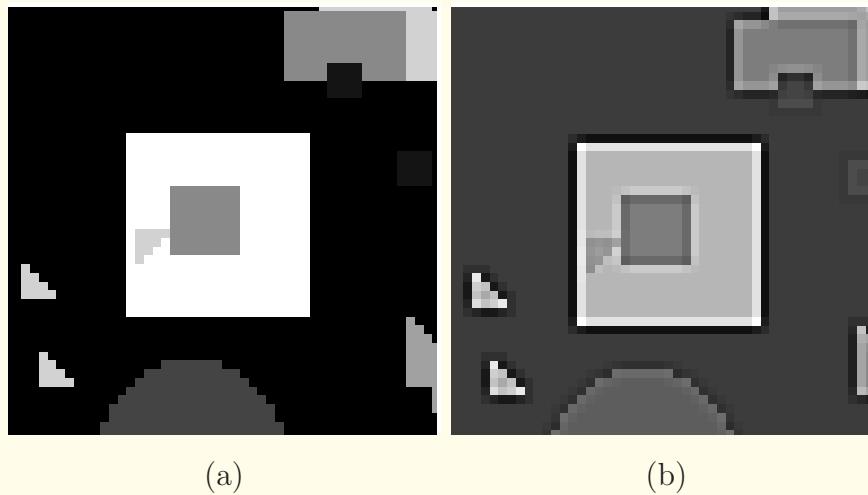


Figure 4.25: Enlarged views of a part of (a) the Shapes test image and (b) the result of unsharp masking; see also Figure 4.24 (a) and (b). Observe the edge-enhancement artifact.



4.6.2 Subtracting Laplacian

Degraded image g modeled as the result of a diffusion process that spreads intensity values over space as a function of time, according to the partial differential equation

$$\frac{\partial g}{\partial t} = \kappa \nabla^2 g, \quad (4.16)$$

where t represents time, $\kappa > 0$ is a constant, and

$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}. \quad (4.17)$$



At $t = 0$, we have $g(x, y, 0) = f(x, y)$, the original image.

At $t = \tau > 0$, the degraded image $g(x, y, \tau)$ is observed.

The degraded image may be expressed in a Taylor series as

$$g(x, y, \tau) = g(x, y, 0) + \tau \frac{\partial g}{\partial t} (x, y, \tau) \quad (4.18)$$

$$- \frac{\tau^2}{2} \frac{\partial^2 g}{\partial t^2} (x, y, \tau) + \dots$$



Ignoring the quadratic and higher-order terms, letting $g(x, y, 0) = f(x, y)$, and using the diffusion model,

$$f_e = g - \kappa \tau \nabla^2 g, \quad (4.19)$$

where f_e represents an approximation to f .

Thus, we have an enhanced image obtained as a weighted subtraction of the given image and its Laplacian (gradient).



Discrete implementation of the Laplacian by a 3×3 mask:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (4.20)$$

Net weight of the coefficients in the Laplacian mask is zero;
therefore, the mask performs a differentiation operation that will
lead to the loss of intensity information
(result in an area of any uniform brightness value will be zero).



Letting the weighting factor $\kappa \tau = 1$ in Equation 4.19, we get the following 3×3 mask known as the subtracting Laplacian:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (4.21)$$

Because the net weight of the mask is equal to unity, the mask retains the local average intensity in the image.



The unsharp masking filter is referred to as the generalized (subtracting) Laplacian by some authors.

The subtracting Laplacian is also an unsharp masking filter.

The subtracting Laplacian also leads to edge enhancement or high-frequency emphasis.



Laplacian: does not maintain intensity information; depiction of the edges (gradient) present in the image.

Subtracting Laplacian: maintains intensity (modified);

provides a sharper image;

could lead to negative pixel values in the enhanced image.

Compression of the larger dynamic range in the enhanced image to a smaller display range could mute the effect of enhancement, and alter the intensity values of parts of the image.

The subtracting Laplacian could also introduce disturbing overshoot and undershoot artifacts around edges.

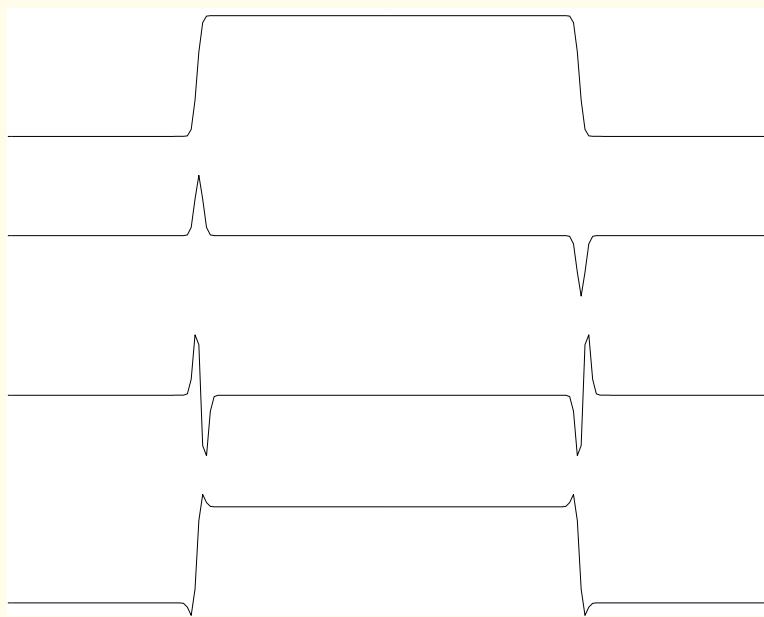


Figure 4.26: Top to bottom: a rectangular pulse signal smoothed with a Gaussian blur function; the first derivative of the signal; the second derivative of the signal; and the result of a filter equivalent to the subtracting Laplacian. The derivatives are shown with enlarged amplitude scales as compared to the original and filtered signals.



4.6.3 *Limitations of fixed operators*

Fixed operators, such as the unsharp-masking and subtracting-Laplacian filters, apply the same mathematical operation at every location over the entire image.

The coefficients and the size of such filters do not vary:

the filters cannot adapt to changes in the nature of the image from one location to another.

Operators with fixed coefficients and fixed spatial scope cannot take nonstationarity and multiscale features into consideration.

Adaptive filters are desirable to address these concerns.



4.7 High-frequency Emphasis

Useful in detecting edges, under the assumption that

high-frequency Fourier spectral components are associated with edges and large changes in the image.

This property follows from the effect of differentiation of an image on its Fourier transform.



The ideal highpass filter:

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \geq D_0 \\ 0 & \text{otherwise.} \end{cases} \quad (4.22)$$

$D(u, v) = \sqrt{u^2 + v^2}$ is the distance of the frequency component at (u, v) from the DC point $(u, v) = (0, 0)$,

with the spectrum being centered,

D_0 is the cutoff frequency, below which all components of the Fourier transform of the given image are set to zero.

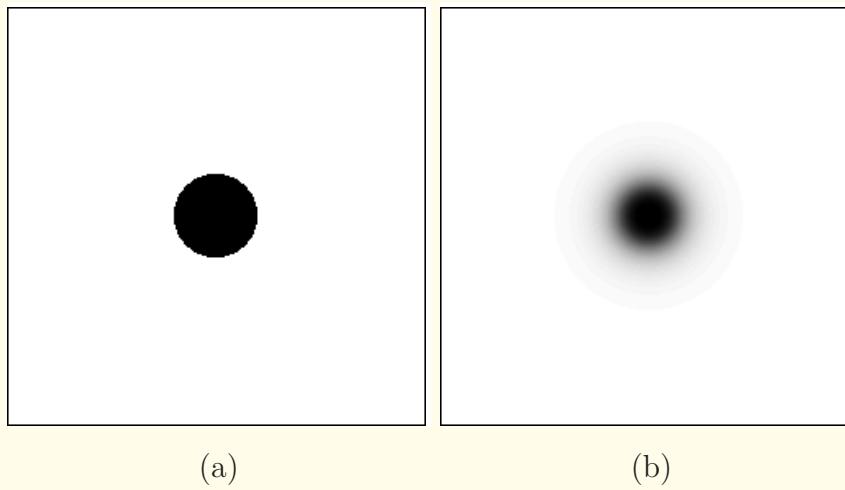


Figure 4.27: (a) The magnitude transfer function of an ideal highpass filter. The cutoff frequency D_0 is 0.2 times the maximum frequency. (b) The magnitude transfer function of a Butterworth highpass filter, with normalized cutoff $D_0 = 0.2$ and order $n = 2$. The $(u, v) = (0, 0)$ point is at the center. Black represents a gain of zero, and white represents a gain of unity. See also Figure 4.28.

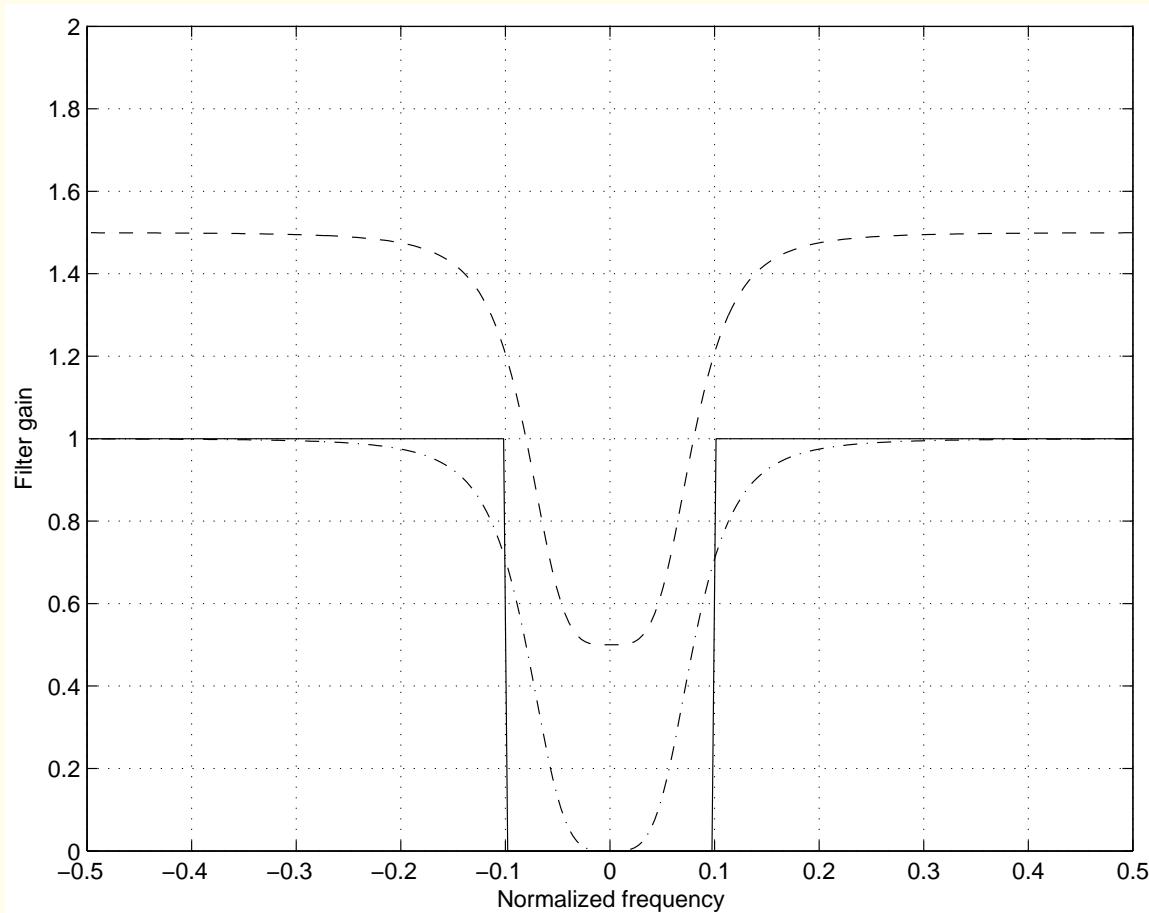


Figure 4.28: Profiles of the magnitude transfer functions of an ideal highpass filter (solid line), a Butterworth highpass filter (dash-dot line, normalized cutoff $D_0 = 0.2$ and order $n = 2$), and a Butterworth high-emphasis filter (dashed line). See also Figure 4.27.



The Butterworth highpass filter:

Prevention of the ringing artifacts encountered with the ideal filter requires that the transition from the stopband to the passband be smooth.

Butterworth filter: monotonic in passband and stopband.

$$H(u, v) = \frac{1}{1 + (\sqrt{2} - 1) \left[\frac{D_0}{D(u, v)} \right]^{2n}}, \quad (4.23)$$



n is the order of the filter,

$$D(u, v) = \sqrt{u^2 + v^2},$$

D_0 is the half-power 2D radial cutoff frequency,

the scale factor in the denominator leads to the gain of the filter being $\frac{1}{\sqrt{2}}$ at $D(u, v) = D_0$.



DC gain of highpass filter = 0 : intensity information removed.

Result depicts only the edges present in the image.

Result will have positive and negative values.

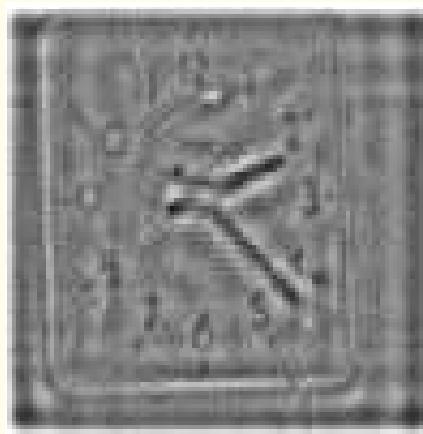
If the enhancement rather than the extraction of edges is desired, it is necessary to maintain the intensity information.

Butterworth high-emphasis filter:

$$H(u, v) = \kappa_1 + \frac{\kappa_2}{1 + (\sqrt{2} - 1) \left[\frac{D_0}{D(u,v)} \right]^{2n}}. \quad (4.24)$$



(a)



(b)



(c)



(d)

Figure 4.29: (a) Clock test image. Result of (b) the ideal highpass filter, display range $[-50, 50]$ out of $[-79, 113]$; (c) the Butterworth highpass filter, display range $[-40, 60]$ out of $[-76, 115]$; and (d) the Butterworth high-emphasis filter, display range $[-40, 160]$ out of $[-76, 204]$.

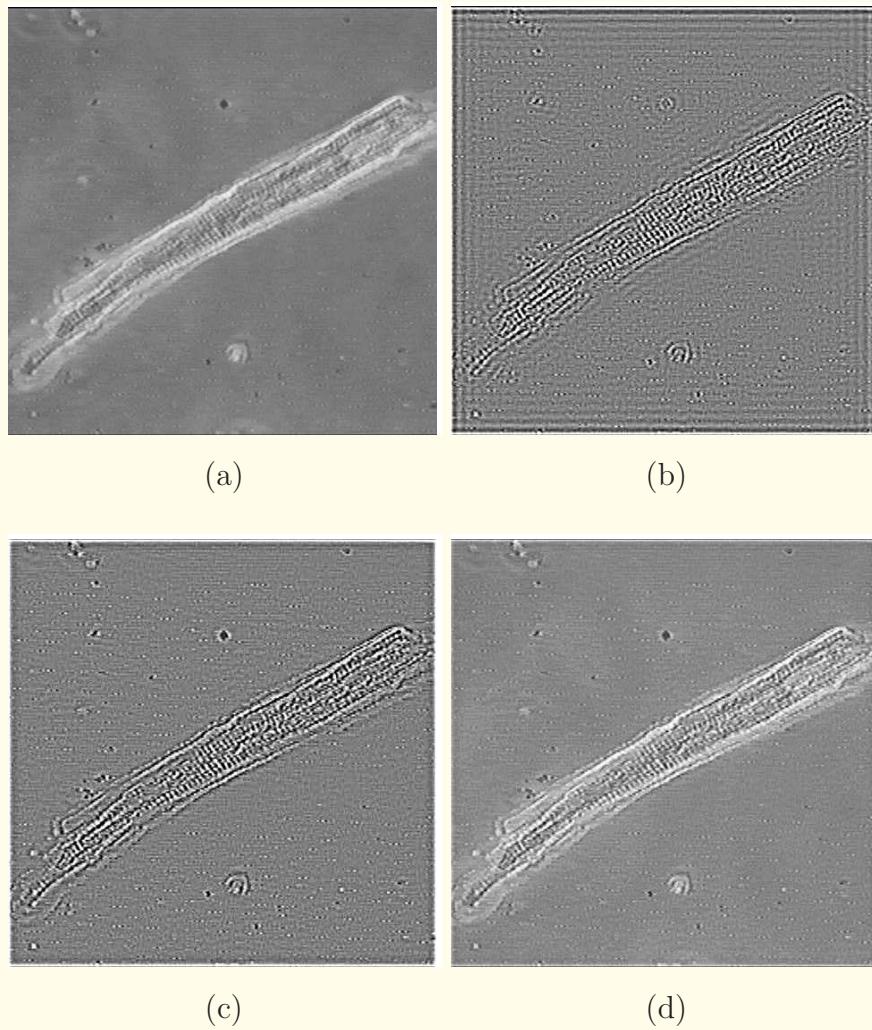


Figure 4.30: (a) Image of a myocyte; the range from the minimum to the maximum of the image has been linearly mapped to the display range $[0, 255]$. Result of (b) the ideal highpass filter, display range $[-20, 20]$ out of $[-60, 65]$; (c) the Butterworth highpass filter, display range $[-20, 20]$ out of $[-61, 61]$; and (d) the Butterworth high-emphasis filter, display range $[-20, 100]$ out of $[-52, 138]$.

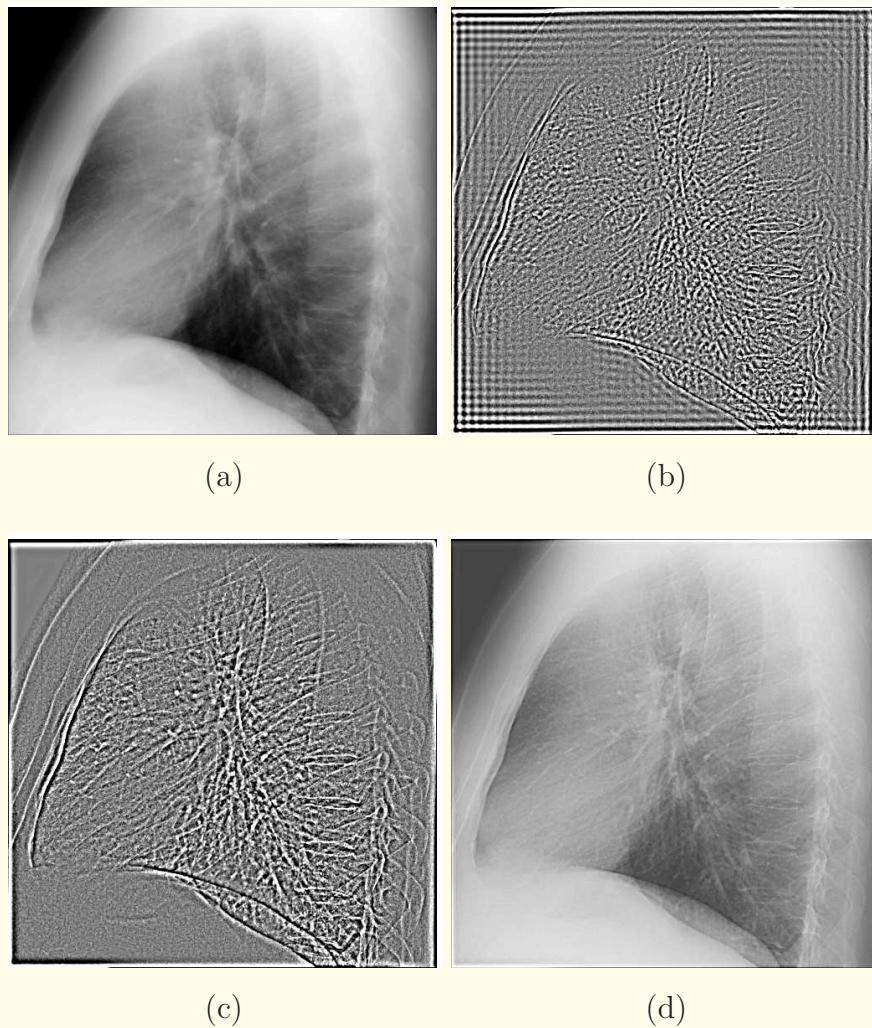


Figure 4.31: (a) Part of a chest X-ray image. Result of (b) the ideal highpass filter, display range $[-5, 5]$ out of $[-74, 91]$; (c) the Butterworth highpass filter, display range $[-5, 5]$ out of $[-78, 95]$; and (d) the Butterworth high-emphasis filter, display range $[-50, 130]$ out of $[-78, 192]$.

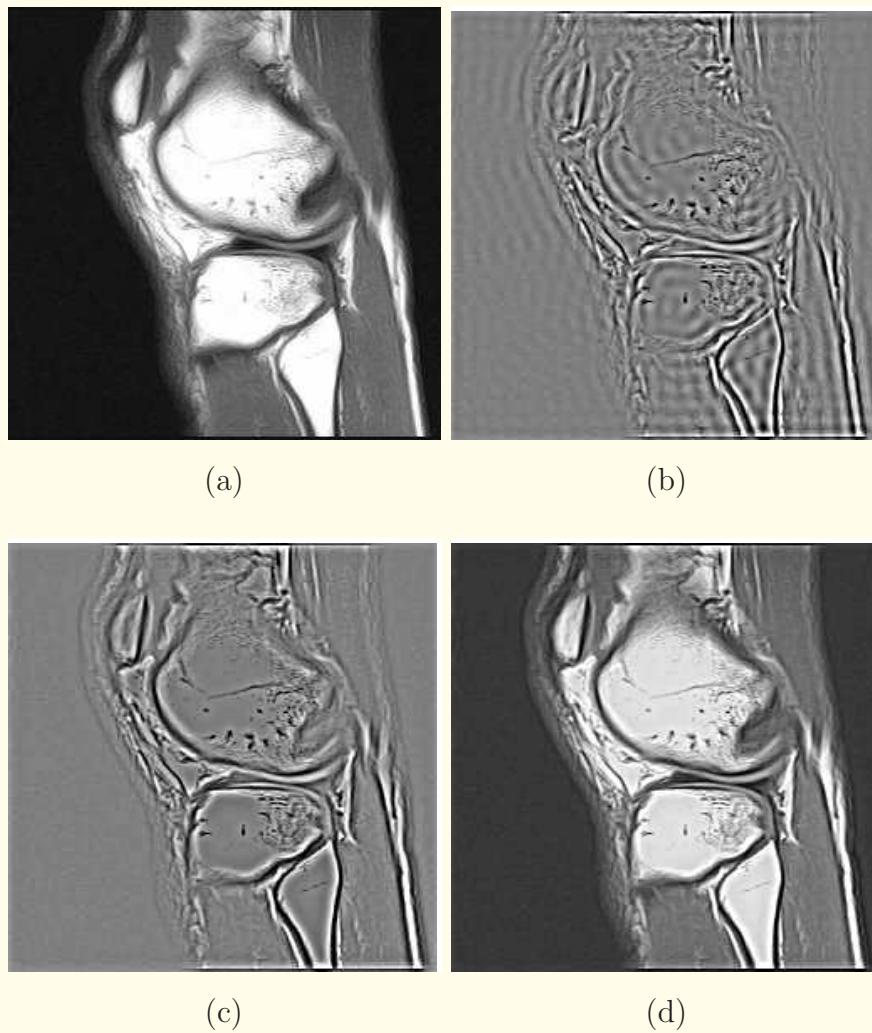


Figure 4.32: (a) MR image of a knee. Result of (b) the ideal highpass filter, display range $[-50, 50]$ out of $[-117, 127]$; (c) the Butterworth highpass filter, display range $[-50, 50]$ out of $[-126, 139]$; and (d) the Butterworth high-emphasis filter, display range $[-30, 150]$ out of $[-78, 267]$.

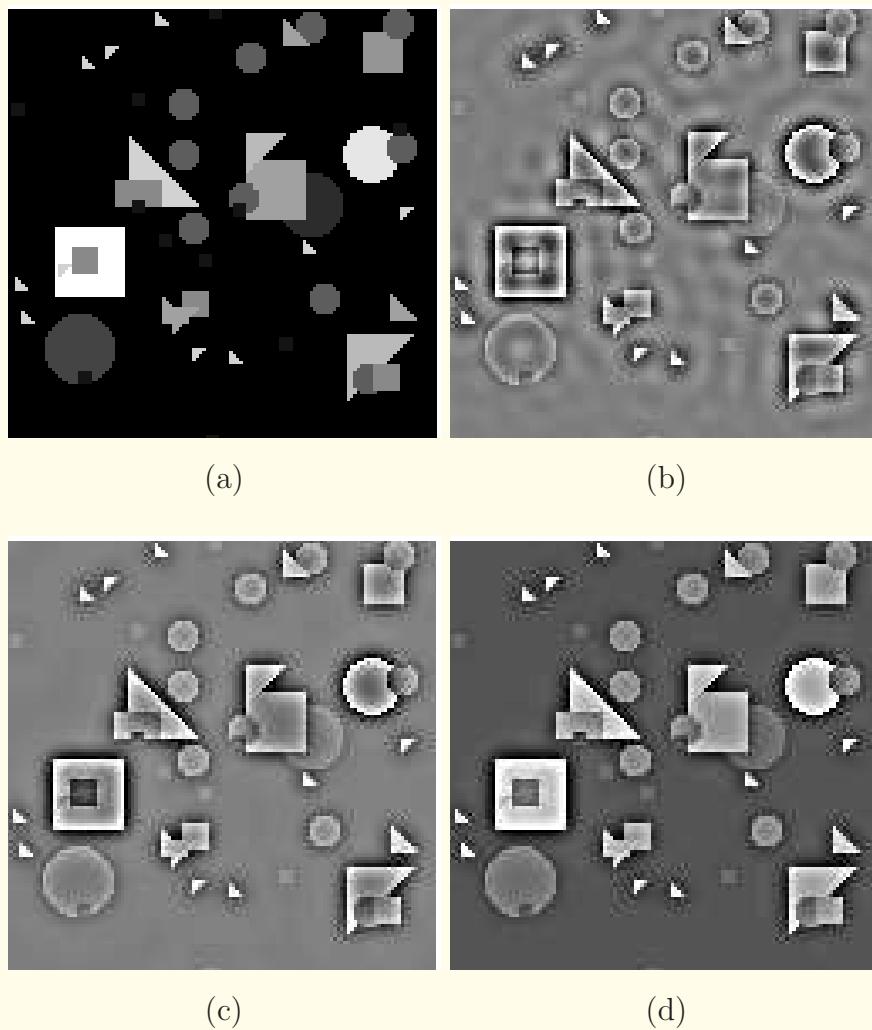


Figure 4.33: (a) Shapes test image. Result of (b) the ideal highpass filter, display range $[-100, 100]$ out of $[-154, 183]$; (c) the Butterworth highpass filter, display range $[-100, 100]$ out of $[-147, 176]$; and (d) the Butterworth high-emphasis filter, display range $[-100, 200]$ out of $[-147, 296]$.



4.8 Homomorphic Filtering for Enhancement

Consider an image that contains the product of two images:

$$g(x, y) = f(x, y) s(x, y).$$

$$G(u, v) = F(u, v) * S(u, v).$$

How would we be able to separate $f(x, y)$ from $s(x, y)$?



Furthermore, suppose we have

$$g(x, y) = h(x, y) * f(x, y).$$

$$G(u, v) = H(u, v) F(u, v).$$

How could we attempt to separate $f(x, y)$ and $h(x, y)$?



4.8.1 *Generalized linear filtering*

Linear filters are well established and understood;

it is attractive to consider extending their application to images that have been combined by operations other than addition,

especially by multiplication and convolution.

Interesting possibility — conversion of the operation combining the images into addition by one or more transforms:

generalized linear filtering,

homomorphic systems.



$$g(x, y) = f(x, y) s(x, y). \quad (4.25)$$

$$\log[g(x, y)] = \log[f(x, y) s(x, y)] = \log[f(x, y)] + \log[s(x, y)], \quad (4.26)$$

$$f(x, y) \neq 0, \quad s(x, y) \neq 0 \quad \forall(x, y).$$

$$G_l(u, v) = F_l(u, v) + S_l(u, v), \quad (4.27)$$

where the subscript l indicates that the Fourier transform has been applied to a log-transformed version of the image.



Assume log transformation has not affected separability of the Fourier components of $f(x, y)$ and $s(x, y)$.

A linear filter (lowpass, highpass, etc.) may now be applied to $G_l(u, v)$ to separate them.

An inverse Fourier transform will yield the filtered image.

An exponential operation will complete the reversal procedure.



Application: image treated as the product of an illumination function and a transmittance or reflectance function.

The homomorphic filter facilitates the separation of the illumination function and correction for nonuniform lighting.

High-emphasis filter leads to simultaneous dynamic range compression and contrast enhancement.

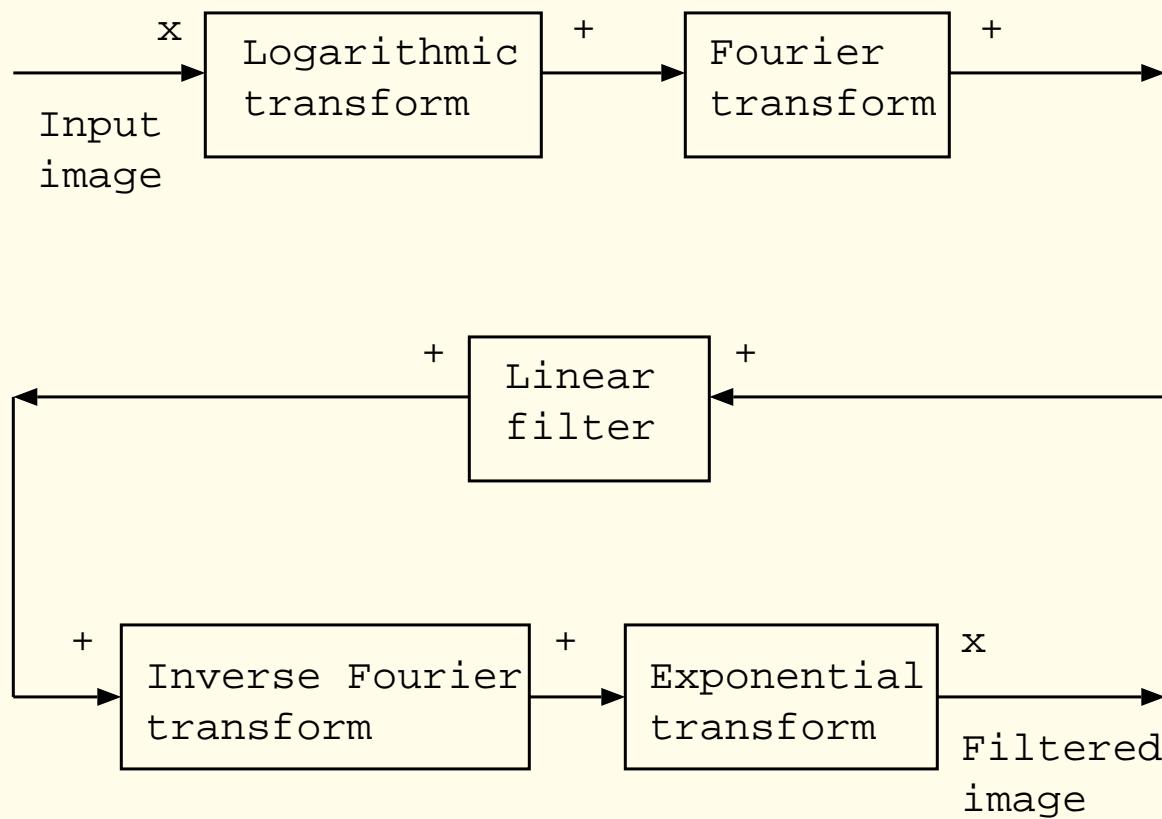


Figure 4.34: Homomorphic filtering for enhancement of images combined by multiplication.

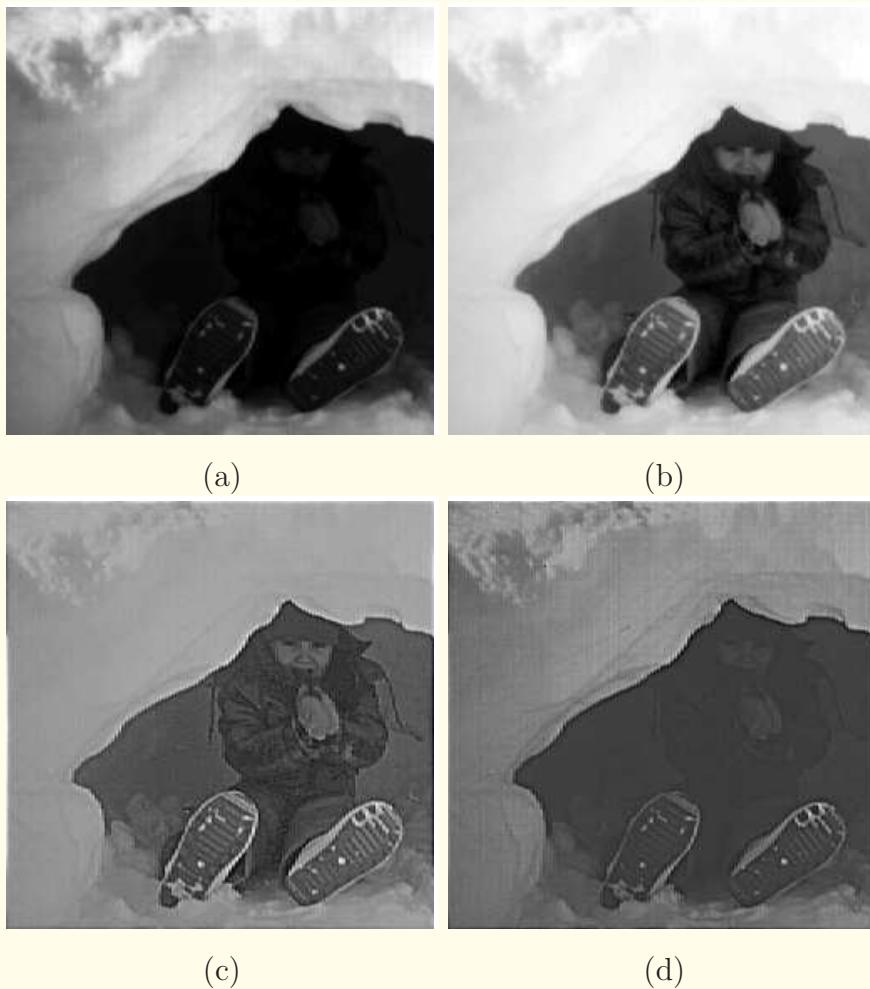


Figure 4.35: (a) Test image of a girl in a snowcave. Result of (b) log transformation; (c) homomorphic filtering including a Butterworth high-emphasis filter; and (d) the Butterworth high-emphasis filter only. The test image in this illustration is of size 256×256 pixels, and is slightly different from that in Figures 4.11 and 4.18; regardless, comparison of the results indicates the advantages of homomorphic filtering. The Butterworth high-emphasis filter used is shown in Figure 4.36. Image courtesy of W.M. Morrow.

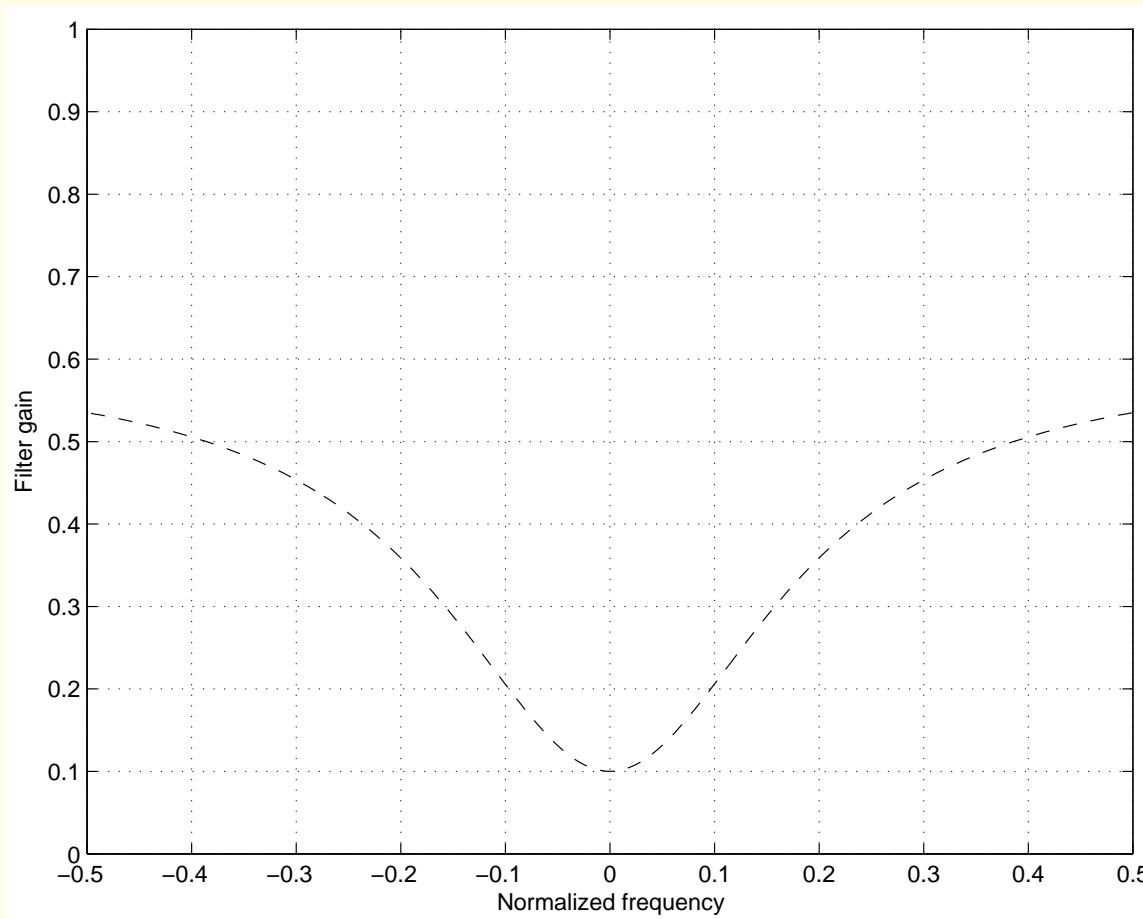


Figure 4.36: Profile of the high-emphasis Butterworth filter used to enhance high-frequency components along with homomorphic filtering as illustrated in Figures 4.34, 4.35, and 4.37.

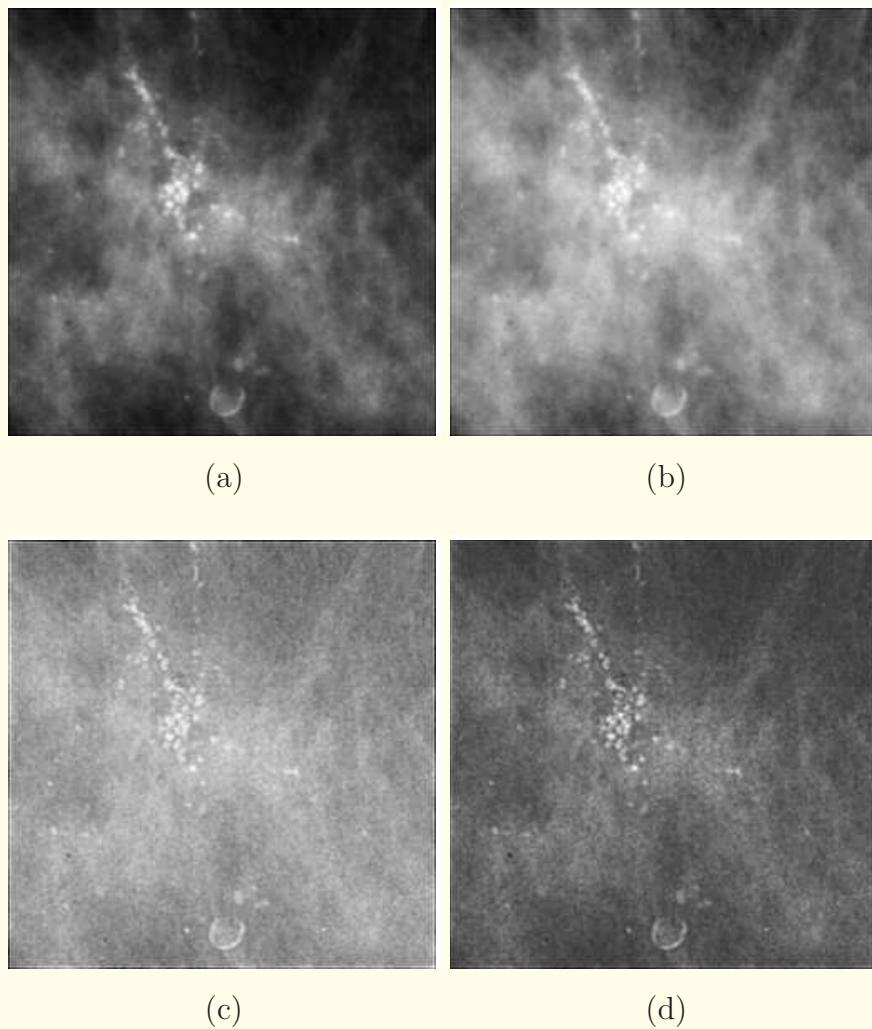


Figure 4.37: (a) Original image of a part of mammogram with malignant calcifications. Result of (b) log transformation; (c) homomorphic filtering including a Butterworth high-emphasis filter; and (d) the Butterworth high-emphasis filter only. See also Figures 4.40 and 4.41.



4.9 Adaptive Contrast Enhancement

Diagnostic features in medical images, such as mammograms, vary widely in size and shape.

Classical image enhancement techniques cannot adapt to the varying characteristics of such features.

The application of a global transform or a fixed operator to an entire image often yields poor results.

It is, therefore, necessary to design methods that can adapt the operation performed or the pixel collection used to derive measures to the local details present in the image.



4.9.1 *Adaptive-neighborhood contrast enhancement*

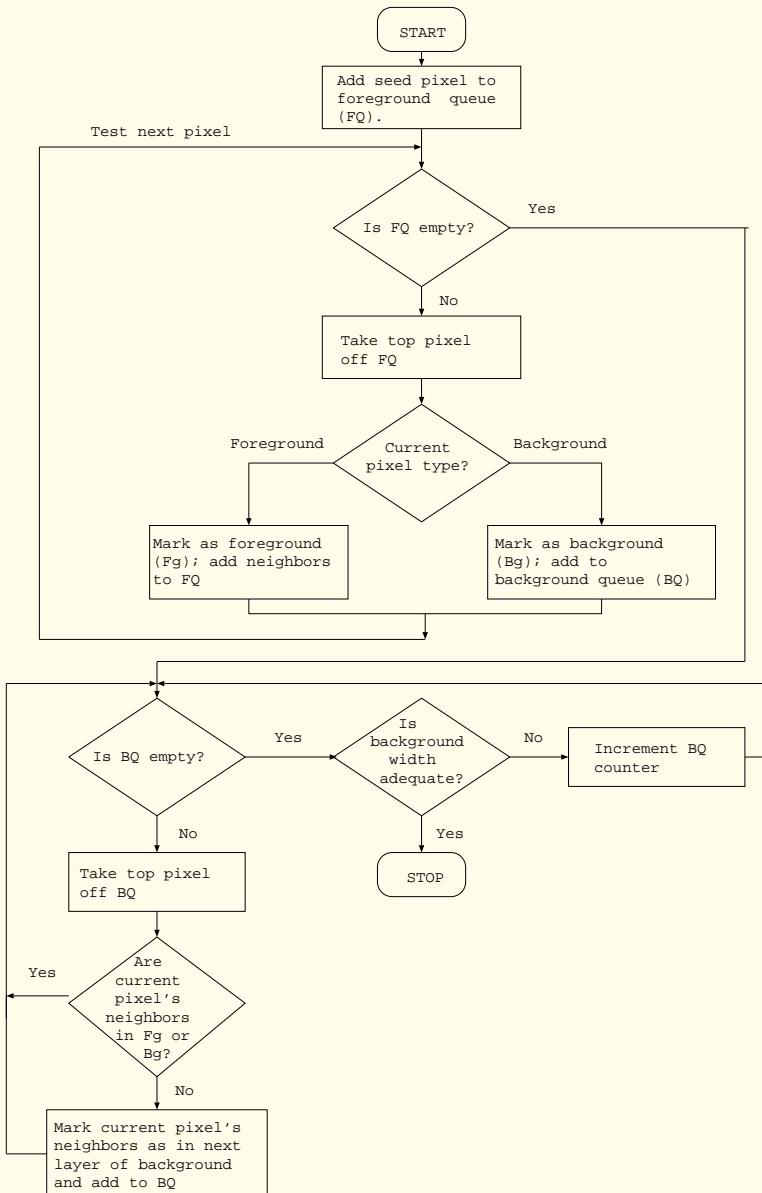


Figure 4.38: Procedure for region growing for adaptive-neighborhood contrast enhancement of mammograms. Reproduced with permission from W.M. Morrow, R.B. Paranjape, R.M. Rangayyan, and J.E.L. Desautels, “Region-based contrast enhancement of mammograms” *IEEE Transactions on Medical Imaging*, 11(3):392–406, 1992. © IEEE.



Adaptive-neighborhood contrast enhancement:

1. The first pixel (or the next unprocessed pixel) in the image is taken as the seed pixel.
2. The immediate neighbors (8-connected pixels) of the seed are checked for inclusion in the region.

Each neighbor pixel is checked to see if its gray-level value is within the specified deviation from the seed pixel's gray level.

The growth tolerance or deviation is specified as

$$\frac{|f(m, n) - \text{seed}|}{\text{seed}} \leq \tau, \quad (4.28)$$

where $f(m, n)$ is the gray-level value of the neighbor pixel being checked for inclusion, and the threshold $\tau = 0.05$.



3. If a neighbor pixel's gray-level value is within the specified deviation, it is added to a queue of foreground pixels that will make up the region being grown.

A pixel is added to the queue only if it has not already been included while processing another connected pixel.

4. A pixel is taken from the start of the foreground queue.

This becomes the current pixel whose 8-connected neighbors are checked against the seed's gray-level according to the tolerance specified, as in Steps 2 and 3 above.



5. If a neighbor pixel's gray-level value is outside the specified gray-level tolerance range, it is marked as a background pixel. A background pixel indicates that the border of the region has been reached at that position.

However, if a neighbor pixel's gray-level value is within the specified deviation, it is added to the foreground.



6. Once all the current pixel's neighbors have been checked, the program goes back to Step 4 to check the connected neighbor pixels of the next pixel in the foreground queue.
7. Steps 4 – 6 are repeated until region growing stops (no more pixels can be added to the foreground region).
8. The borders of the foreground are expanded in all directions by a prespecified number of pixels to obtain a background that is molded to the shape of the foreground.

The foreground and background regions together form the adaptive neighborhood of the seed pixel that was used to start the region-growing procedure.



9. The contrast of the region is computed and enhanced.

Gray level of seed pixel modified as per Equation 4.30.

All pixels in the foreground region having the same gray-level value as the seed, referred to as the redundant seed pixels, are also modified to the same value as for the seed pixel.

10. Steps 1 – 9 are executed until all the pixels in the image have been processed.



Parameters for seed-fill region growing:

Growth tolerance τ .

Seed-fill region growing results in regions having contrast greater (in magnitude) than a certain minimum contrast, C_{\min} .

It is desired that this minimum contrast be independent of a region's gray level, so that the results of enhancement will be independent of a multiplicative transformation of the image.

A region with the minimum positive contrast C_{\min} will have a mean foreground value of f and a mean background value of $(1 - \tau)f$.



$$C_{\min} = \frac{f - (1 - \tau)f}{f + (1 - \tau)f} = \frac{\tau}{2 - \tau} \approx \frac{\tau}{2}. \quad (4.29)$$

The contrast C_{\min} is thus independent of the foreground gray level or the background gray level, and depends only upon the region-growing tolerance parameter τ .



Weber's ratio of 2% for a just-noticeable feature suggests that the growth tolerance should be about 4%,

in order to grow regions that are barely noticeable prior to enhancement (and are subsequently enhanced to a contrast above the Weber ratio).

A lower bound on τ may be established empirically, or, depending upon the class of images being enhanced, through an analysis of the noise present in the images.



Contrast enhancement:

$$C = \frac{f - b}{f + b}.$$

The contrast of a region may be increased by changing f or b .

Replacing C with an increased contrast C_e and rearranging:

$$f_e = b \frac{1 + C_e}{1 - C_e}, \quad (4.30)$$

where f_e is the new foreground value.

Only the seed pixel and the redundant seed pixels in the foreground are modified to the value f_e .

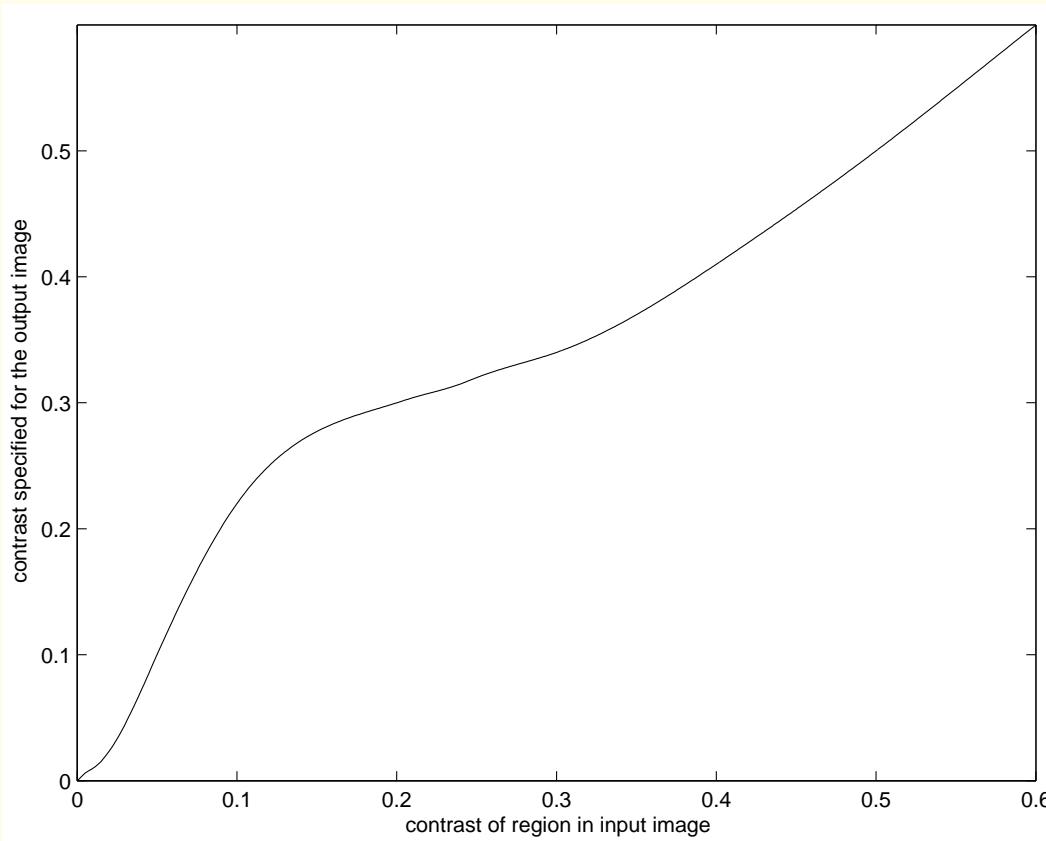


Figure 4.39: An empirical relationship between the contrast C of an adaptive neighborhood and the increased contrast C_e for enhancement of mammograms. $C_e = C$ for $C \geq 0.5$.

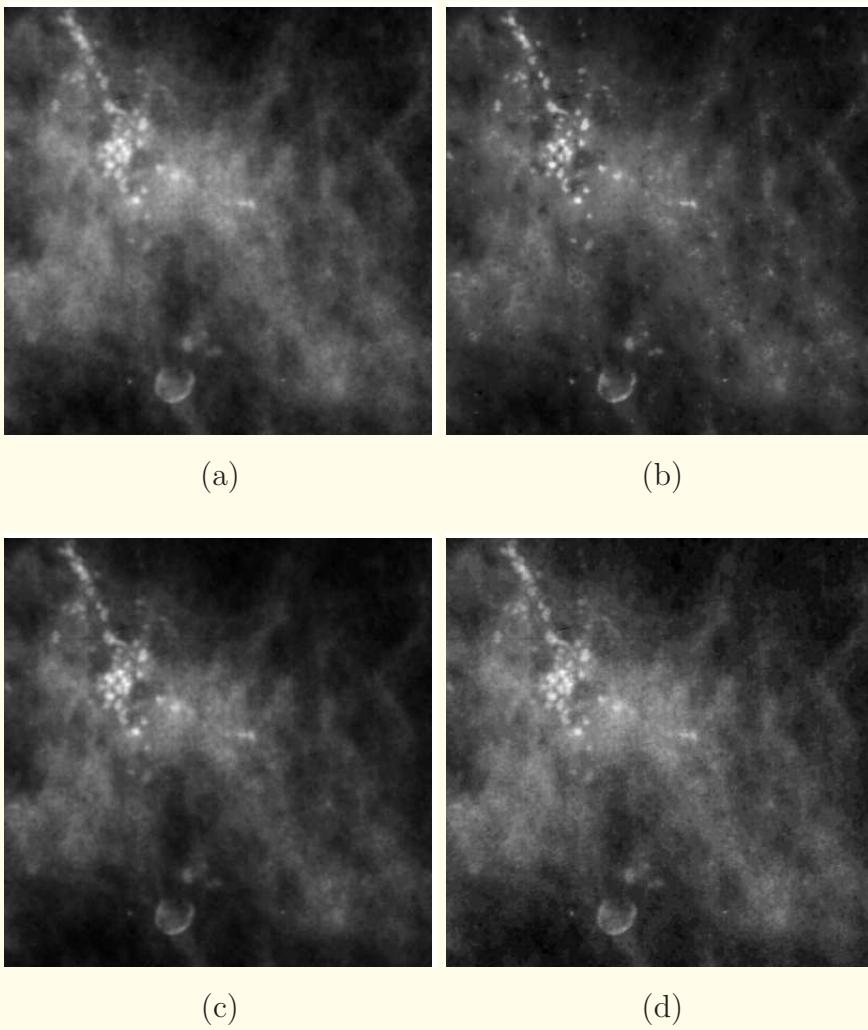


Figure 4.40: (a) Part of a mammogram with a cluster of calcifications, true size $43 \times 43 \text{ mm}$. Results of enhancement by (b) adaptive-neighborhood contrast enhancement; (c) gamma correction; and (d) unsharp masking. See also Figures 4.37 and 4.41. Reproduced with permission from W.M. Morrow, R.B. Paranjape, R.M. Rangayyan, and J.E.L. Desautels, “Region-based contrast enhancement of mammograms” *IEEE Transactions on Medical Imaging*, 11(3):392–406, 1992. © IEEE.

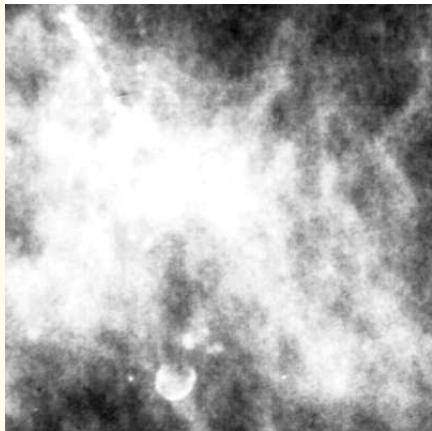
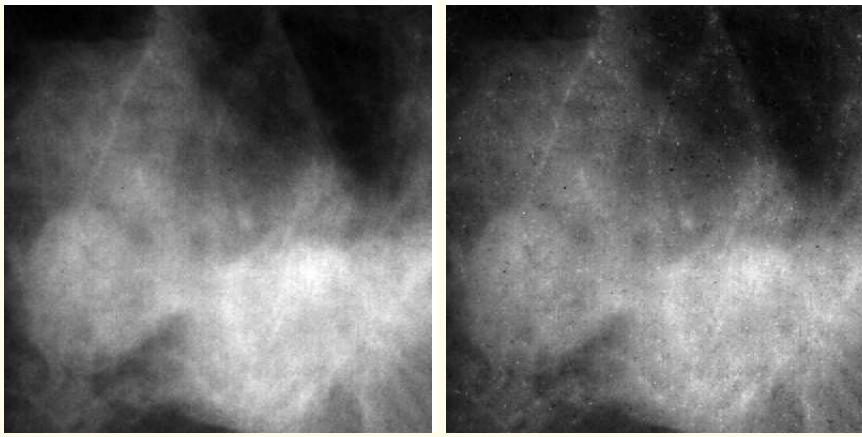


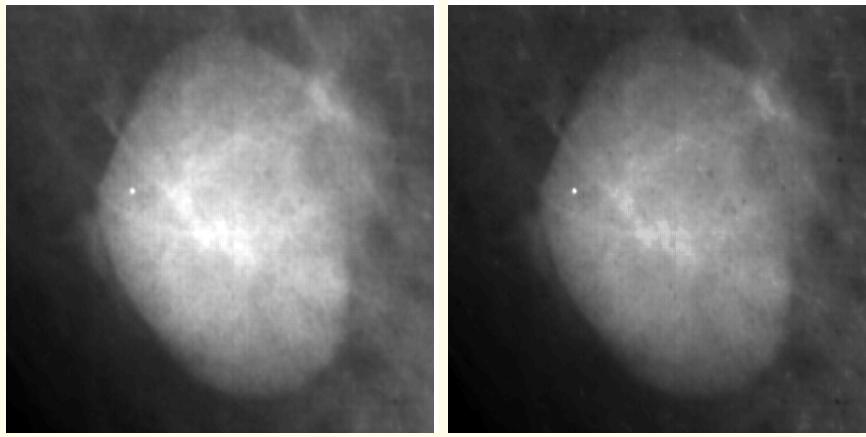
Figure 4.41: Result of enhancement of the image in Figure 4.40 (a) by global histogram equalization applied to the entire image. See also Figures 4.37 and 4.40. Reproduced with permission from W.M. Morrow, R.B. Paranjape, R.M. Rangayyan, and J.E.L. Desautels, “Region-based contrast enhancement of mammograms,” *IEEE Transactions on Medical Imaging*, 11(3):392–406, 1992. © IEEE.



(a)

(b)

Figure 4.42: (a) Part of a mammogram with dense masses, true size $43 \times 43 \text{ mm}$. (b) Result of enhancement by adaptive-neighborhood contrast enhancement. Reproduced with permission from W.M. Morrow, R.B. Paranjape, R.M. Rangayyan, and J.E.L. Desautels, “Region-based contrast enhancement of mammograms,” *IEEE Transactions on Medical Imaging*, 11(3):392–406, 1992. © IEEE.



(a)

(b)

Figure 4.43: (a) Part of a mammogram with a benign cyst, true size $43 \times 43 \text{ mm}$. (b) Result of enhancement by adaptive-neighborhood contrast enhancement. Reproduced with permission from W.M. Morrow, R.B. Paranjape, R.M. Rangayyan, and J.E.L. Desautels, “Region-based contrast enhancement of mammograms,” *IEEE Transactions on Medical Imaging*, 11(3):392–406, 1992. © IEEE.



4.10 Objective Assessment of Contrast Enhancement

Spread of the contrast histogram:

For a distribution of contrast values c_i , quantized so that there are N bins over the range $[-1, 1]$, the second moment M_2 is

$$M_2 = \sum_{i=1}^N c_i^2 p(c_i), \quad (4.31)$$

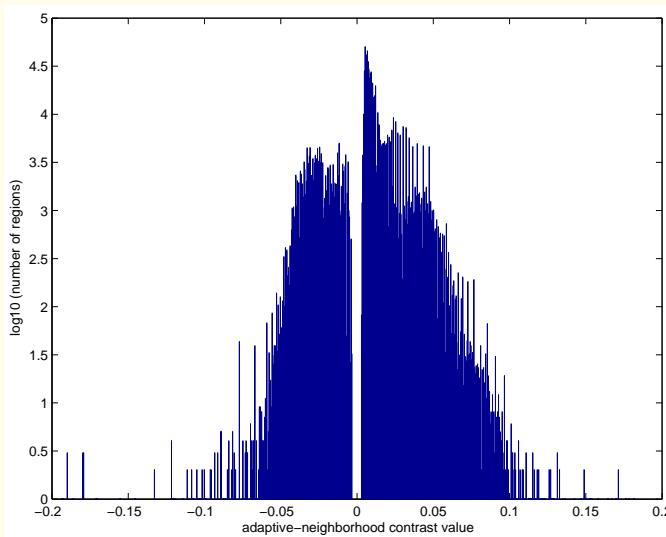
where $p(c_i)$ is the normalized number of seed pixels that lead to the growth of a region with contrast c_i .



Low-contrast image: narrow contrast histogram: low M_2 .

High contrast: broader contrast histogram: greater M_2 .

Image contrast needs to be recomputed after the entire image has been enhanced.



(a)

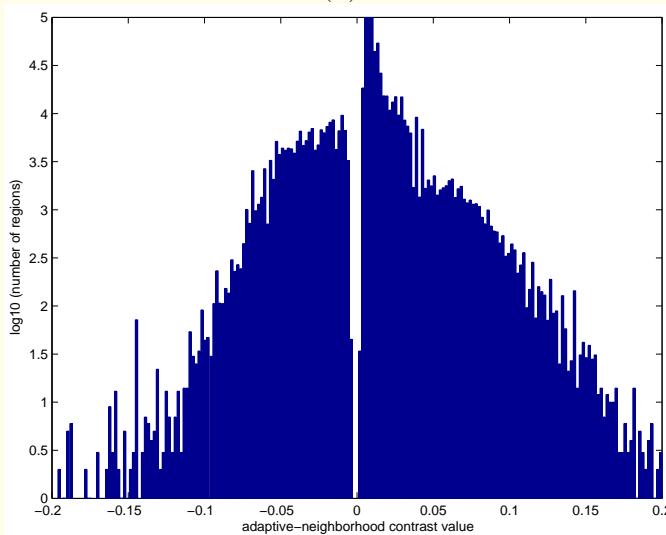
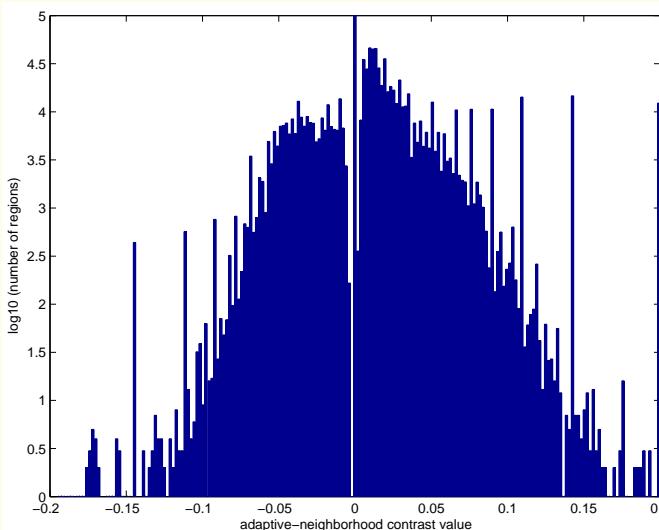
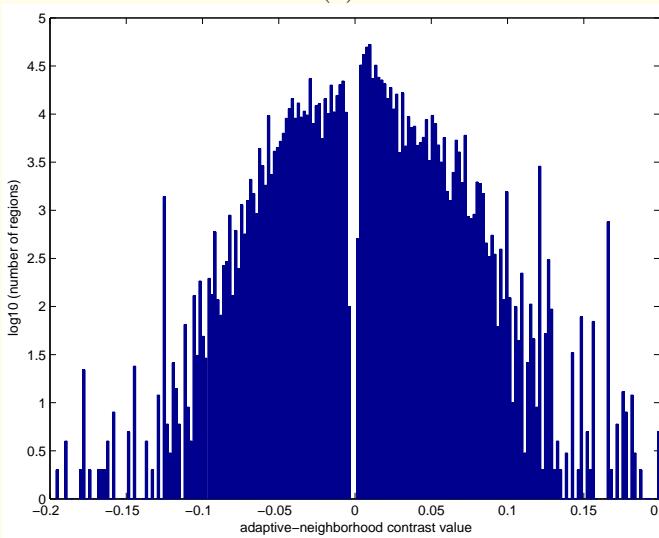


Figure 4.44 (b)



(c)



(d)

Figure 4.44: Contrast histograms of the full mammograms corresponding to the images in Figure 4.40. (a) Original, $M_2 = 3.71 \times 10^{-4}$; (b) adaptive-neighborhood contrast enhancement, $M_2 = 6.17 \times 10^{-4}$; (c) gamma correction, $M_2 = 3.2 \times 10^{-4}$; and (d) unsharp masking, $M_2 = 4.4 \times 10^{-4}$. Reproduced with permission from W.M. Morrow, R.B. Paranjape, R.M. Rangayyan, and J.E.L. Desautels, “Region-based contrast enhancement of mammograms,” *IEEE Transactions on Medical Imaging*, 11(3):392–406, 1992. © IEEE.



