Chapter 3

IMAGE ENHANCEMENT

Concepts Introduced in this Chapter

- Measures of image resolution and contrast;
- The modulation transfer function;
- Signal-to-noise ratio;
- Contrast and resolution enhancement;
- Noise reduction and edge enhancement by linear filtering;
- Edge-preserving smoothing by median filtering, diffusion filtering, various approximations of diffusion filtering and Bayesian image restoration.

Introduction

- Two reasons exist for applying an image enhancement technique:
- Enhancement can increase perceptibility of objects in an image to the human observer or It may be needed as a preprocessing step for subsequent automatic image analysis.
- Enhancement methods differ for the two purposes. An enhancement method requires a criterion by which its success can be judged. This will be a definition of image quality, since improving quality is the goal of such method.
- Different enhancement techniques will be presented covering methods for contrast enhancement, for the enhancement of edges, and for noise reduction.
- In medical imaging, image enhancement essentially enhances contrast by reducing any artefacts or noise in the image or by emphasizing differences between objects.
- Quantitative measures of image quality will help as they describe aspects of an image that are relevant to human or computer analysis independent of an observer.
- Improvement of quality measures is then evidence for success of an enhancement procedure. These include image content, observation task, visualization quality, and performance of the observer.

Measures of Image Quality

1) Spatial Resolution:

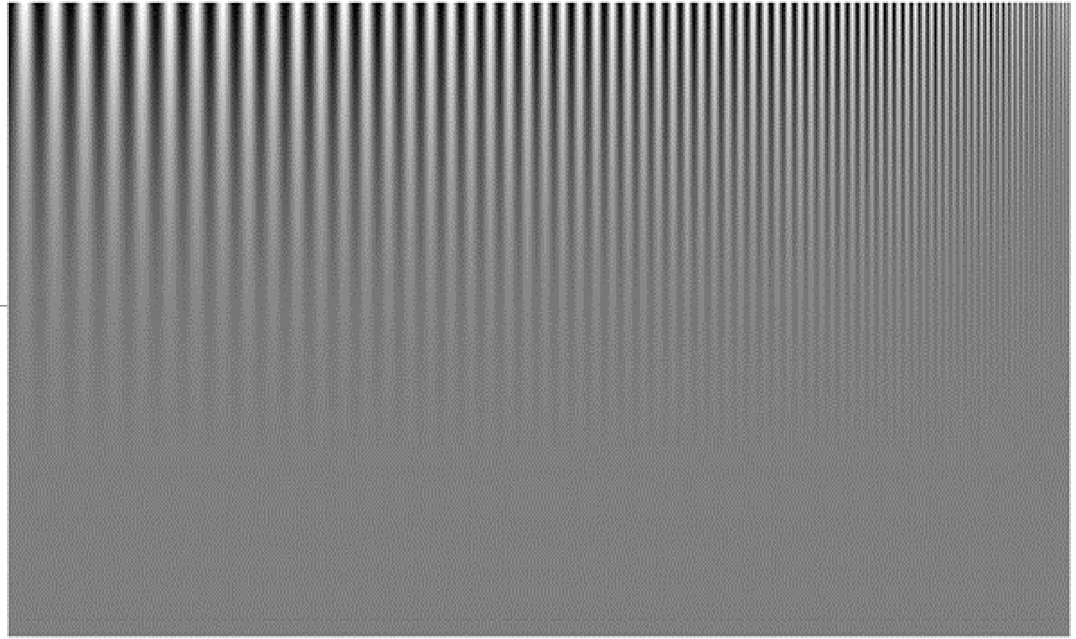
- Spatial and contrast resolution -characterize images , determine the smallest structure that can be represented in a digital image. These two measures are easily computable and relevant to digital image processing.
- Structures can only be analyzed (delineated, measured etc.), if they appear in the image. Spatial resolution describes this directly since the sampling theorem states that no detail with frequency less than twice the sampling distance can be represented without aliasing.
- The contrast resolution is an indirect measure of perceptibility of structures.
- The number of intensity levels has an influence on the likelihood with which two neighboring structures with similar but not equal appearance will be represented by different intensities.

Resolution

- Technical measures of resolution do not relate directly to the ability of humans to recognize a structure.
- perceived resolution, as opposed to technical resolution, cannot easily be reduced to a single cause such as spatial resolution.
- Perceived resolution may be measured experimentally by treating the human visual system as a black box system with images as input and recognized objects determining resolution as output.
- The same kind of measure is also used when loss of resolution by transfer of information through a technical system shall be documented (such as creating a radiograph from a scene).
- The quantity that is measured is called **line pairs per millimeter (lpmm),** which refers to the thinnest pair of parallel black and white lines that can be differentiated.
- A sequence of parallel pairs of black and white lines with decreasing line thickness is displayed. **Apparent resolution is 1/x lpmm if the thickness of the thinnest line pair is x mm**. The measure is proportional to the frequency, but it is a more figurative expression and easier to understand.

Perceived resolution

- Perceived resolution by a human is not independent of contrast.
- **An object stands out less against the background, if the intensity difference between object and background is low.**
- Perceived resolution may sometimes be even higher than technical resolution because decreasing contrast may be interpreted as PVE due to the subvoxel size of the object.
- For instance, vessels that are smaller than the voxel size are visible in contrast-enhanced MR angiography because of the PVE.



A test pattern for determining perceived resolution in line pairs per millimeter (lpmm). The number of line pairs per millimeter increases from left to right while the contrast decreases from top to bottom. It can be seen that perceived resolution depends on the contrast in the image and also that this relationship is non-linear

Definition of Contrast

- Determining contrast requires knowledge about what is an object and what is background.
- Since this is unknown prior to analysis, a number of measures for calculating image contrast exist that make implicit assumptions about image content.
- Examples for object-independent contrast measures are **global contrast, global variance, entropy, and contrast from the co-occurrence matrix.**

The Michelson contrast $C_{Michelson}$ measures the utilization of the luminance range. The smallest displayed luminance is f_{min} and the highest displayed luminance is f_{max}. Michelson contrast is then:

$$C_{\mathrm{Michelson}} = \frac{f_{\mathrm{max}} - f_{\mathrm{min}}}{f_{\mathrm{max}} + f_{\mathrm{min}}}.$$

Global contrast C_{global} extends this concept by also assuming a maximum (displayable) luminance l_{max}

$$C_{\mathrm{global}} = \frac{f_{\mathrm{max}} - f_{\mathrm{min}}}{l_{\mathrm{max}} + l_{\mathrm{min}}}.$$

- The measure assumes a simple image in which the number of foreground pixels approximately equals that of the background pixels.
- Global contrast ranges from 0 to 1. It is 1.0 if the full range of intensity values is used and less than 1.0 otherwise. However, it does not account for the distribution of intensities in the image.
- An image could be highly underexposed with most of the pixels having intensities below some low threshold but having just one pixel with value I_{max}, possibly caused by an artefact.
- A somewhat better approach for measuring global contrast is the root-mean square (rms) contrast . Given an image f(x, y) with intensities l(x, y), the expected value of i is,

$$\bar{l} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} l(i,j),$$

- The measure takes all pixels into account instead of just the pixels with maximum and minimum intensity value.
- C_{rms} does not differentiate well between different intensity distributions.

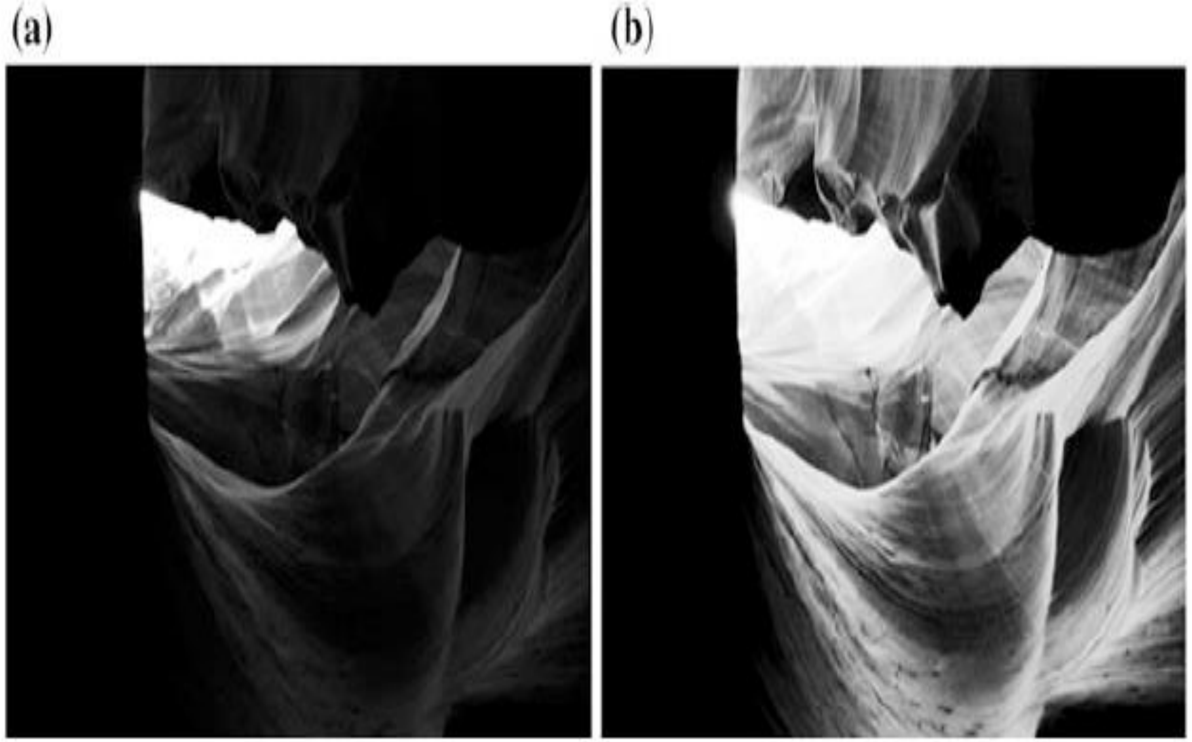


Fig. 4.2 Two images have the same global contrast C_{global}, while their local rms contrast C_{rms} differs by a factor of three [C_{rms} = 0.006 for (**a**) and C_{rms} = 0.018 for (**b**)]

rms contrast is,

$$C_{rms}(f) = \sqrt{\frac{1}{MN-1} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (l(i,j) - \bar{l})^2}.$$

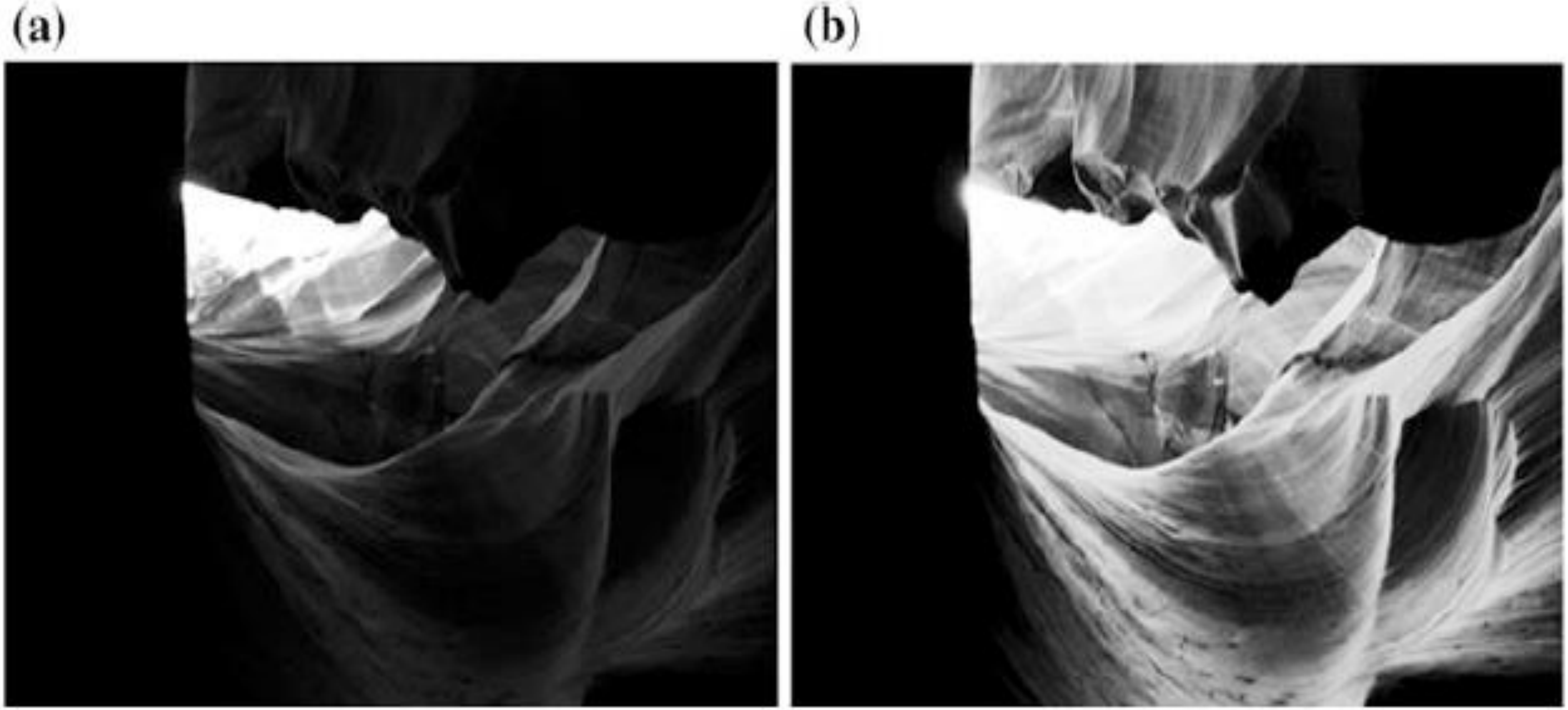


Fig. 4.2 Two images have the same global contrast C_{global}, while their local rms contrast C_{rms} differs by a factor of three [C_{rms} = 0.006 for (**a**) and C_{rms} = 0.018 for (**b**)]

Entropy

- Entropy as contrast measure includes histogram characteristics into the measure. It is computed from the normalized histogram of image intensities. A histogram H(l) of an image l(x, y) gives the frequency of occurrence for each intensity value.
- A normalized histogram $H_{norm}(l)$ is computed from H(l) by

$$H_{\text{norm}}(l) = \frac{H(l)}{\sum_{k=l_{\min}}^{l_{\max}} H(k)}.$$

- It gives the probability of l to appear in an image. If $H_{norm}(20) = 0.05$, the probability is 0.05 that the gray value of a randomly picked pixel is 20. Entropy is computed from H_{norm}.
- It is being used in information theory for determining the average information capacity of a pixel. Information capacity is defined assuming that information I(l) of a pixel with intensity l is inversely proportional to the probability of its occurrence.
- Entropy still does not account for the fact that contrast should measure intensity differences between some foreground and background object

$$I(l) = \left(H_{\text{norm}}(l)\right)^{-1}.$$

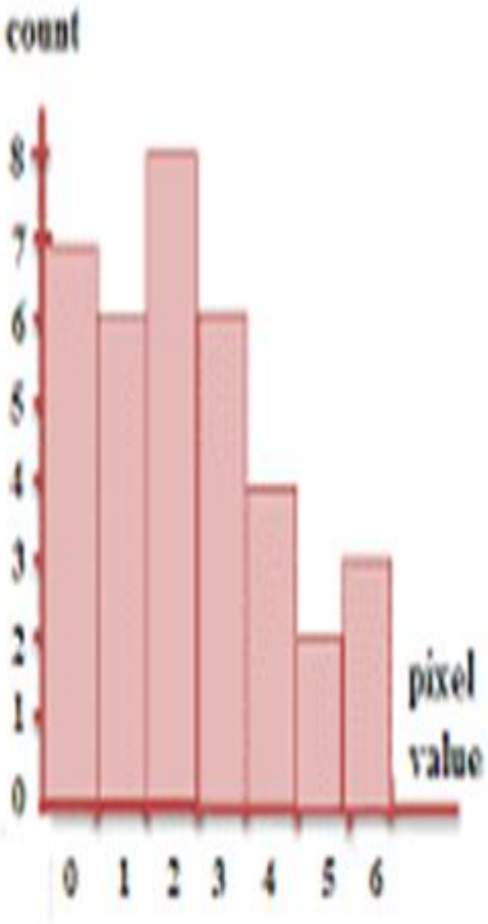
Image Histogram

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

Image 6*6

Pixel value	count
0	7
1	6
2	8
3	6
4	4
5	2
6	3
Total	36

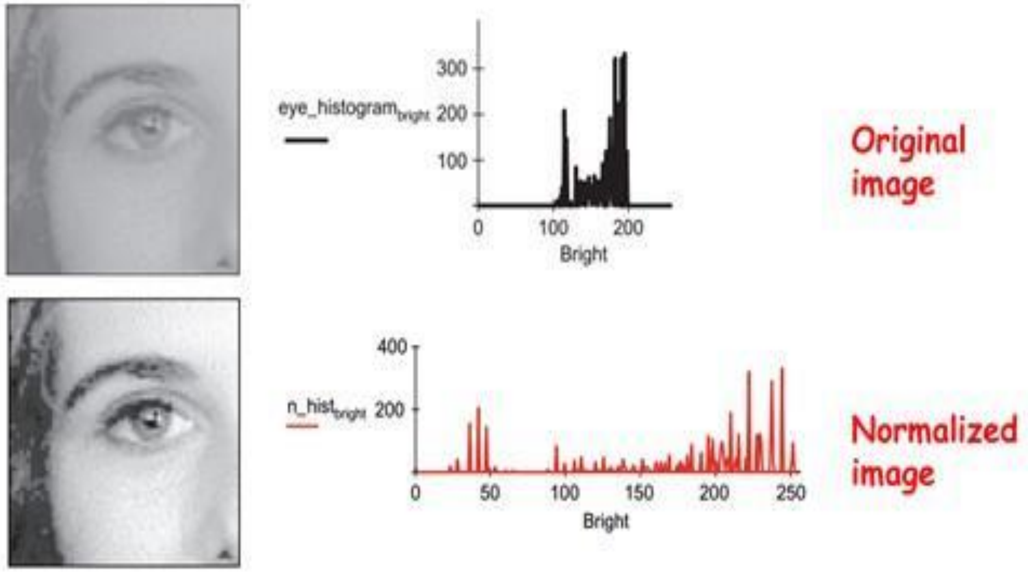
Histogram



Plot of the histogram

- Histogram Normalization
  - Stretch and shift the original histogram
  - Cover all the 256 available levels

$$\mathbf{N}_{x,y} = \frac{\mathbf{N}_{\max} - \mathbf{N}_{\min}}{\mathbf{O}_{\max} - \mathbf{O}_{\min}} \times (\mathbf{O}_{x,y} - \mathbf{O}_{\min}) + \mathbf{N}_{\min} \quad \forall x, y \in 1, N$$



If information is stored in a binary number, then the number of required digits would be

$$\mathrm{SI}(l) = \log_2 \frac{1}{H_{\mathrm{norm}}(l)} = -\log_2 H_{\mathrm{norm}}(l). \tag{4.7}$$

Distribution of intensities of all pixels in the histogram can be used to compute the lowest number of digits needed to encode the image

$$\mathrm{SI}_{\mathrm{total}}(H) = -\sum_{k=I_{\min}}^{I_{\max}} H_{\mathrm{norm}}(k) \log_2 H_{\mathrm{norm}}(k). \tag{4.8}$$

The entropy C_{entropy} is then the average signal length needed (see Fig. 4.3 for an example):

$$C_{\mathrm{entropy}}(H) = -\frac{1}{MN} \sum_{k=I_{\min}}^{I_{\max}} H_{\mathrm{norm}}(k) \log_2 H_{\mathrm{norm}}(k), \tag{4.9}$$

where MN is the number of pixels in f.

gray-level co-occurrence matrix (GLCM)

- This can be computed using a gray-level co-occurrence matrix (GLCM). Co-occurrence calculates the normalized rates of co-occurring intensity values in a given neighborhood.
- The neighborhood is defined by the distance and direction between the two pixels. co-occurrence $C_{a,d}$ is a two-dimensional function of intensities l1 and l2. Ca,d(l1,l2) is the probability with which pixels with intensities l1 and l2 occur such that pixel l1 and l2 are d units apart at an angle of a with the x-axis.
- Co-occurrence matrices can be computed with different distances and different directions representing intensity changes between structures at different angles and with different sharpness at the edge.
- For measuring contrast in a given image, co-occurrence is computed for a fixed distance (e.g., d = 1 pixel) and for arbitrary angles. Cd(l1, l2) is then the co-occurrence of pixels with gray levels l1 and l2 at distance d with arbitrary angle .For d = 1, this would be the four pixels of the 4-neighbourhood. Contrast CGLCM is then defined as,

$$C_{GLCM}(i,j) = \frac{1}{I_{max}^2} \sum_{i=0}^{I_{max}} \sum_{j=0}^{I_{max}} C_d(i,j)\left(1+(i-j)^2\right) - 1.$$

- Contrast thus weights co-occurrences of two intensities by the difference between the two. Higher differences indicating edges receive higher weights.

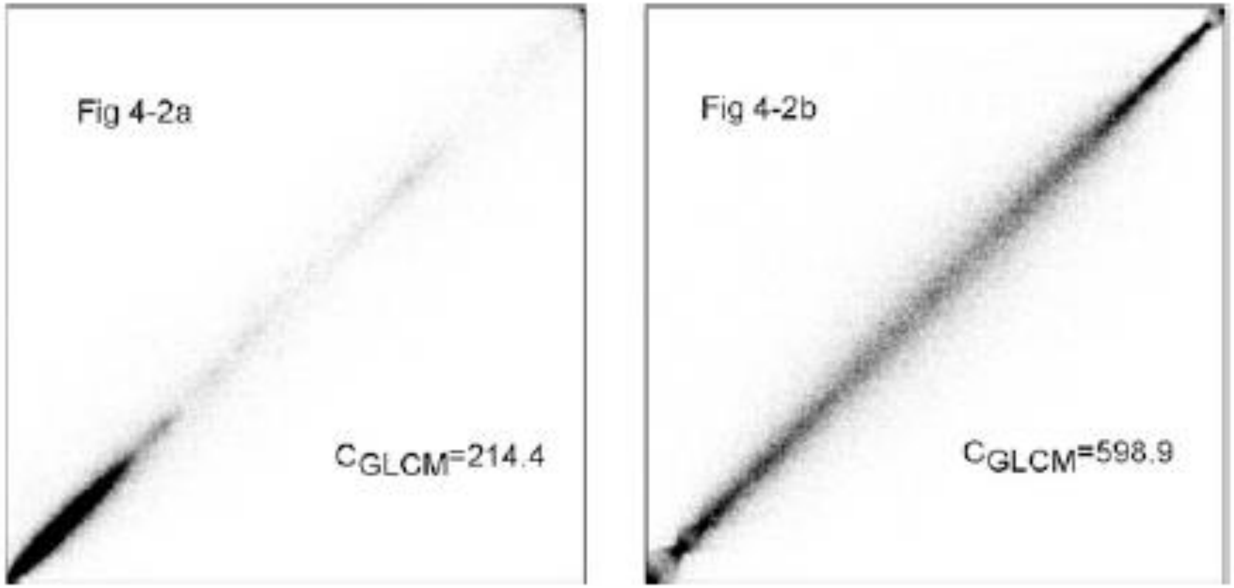


Fig. 4.4 Co-occurrence matrices for the two pictures in Fig. 4.2 and entropy-based contrast measure C_{GLCM}

The Modulation Transfer Function

- To a large extent, improvement of contrast and resolution are constrained by technical parameters of an imaging modality.
- Artefacts, noise, and approximations in the reconstruction may reduce contrast in the original measurement due to reconstruction.
- The degradation that an image suffers through reconstruction, transfer or any other process that changes contrast, is described by the modulation transfer function (MTF).
- **An MTF is a function of frequency (or of resolution in lpmm), which describes the extent of signal loss (damping) with respect to frequency.**
- It is used, for instance, to describe the loss of contrast when creating an X-ray image.
- It can also be used to describe the discrimination performance of human vision with respect to varying contrast.
- contrast at low frequency needs to be higher for recognizing an object than in midrange.

Signal-to-Noise Ratio (SNR)

- Noise in an image is another factor limiting the perceptibility of objects in an image.
- Noise is an unwanted, image-corrupting influence.
- Noise n(i, j) in an image is usually described as a random fluctuation of intensities with zero mean.
- If noise is assumed to be normally distributed with zero mean, variance $\sigma^2(n)$ or standard deviation $\sigma(n)$ characterizes the noise level.
- Object detection depends on the ratio of object-background contrast to noise variance. The former is called difference signal between object and background, and it is related to noise in the signal-to-noise ratio (SNR):

$$\mathrm{SNR}(f) = S(f)/\sigma(n).$$

- In the simplest case, signal S(f) is defined to be the largest intensity fmax (peak SNR) or the average intensity E(f).

Signal-to-Noise Ratio (SNR)

- A common measure, which is measured in dB, is given by the logarithm of the ratio of signal and noise variance is,

$$\mathrm{SNR_{dB}} = 10 \cdot \log_{10}\left[\frac{1}{MN}\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(f(i,j)-\bar{f}\right)^2}{\sigma^2}\right].$$

- Increase in SNR may indicate enhancement of the image. If SNR shall be used as an absolute quantity to determine the perceptibility of objects, the difference signal has to reflect the difference between object and background intensity.
- None of the quantities listed covers the dependency of recognizing objects based on their size, shape, sharpness of edges, and texture as it would require prior detection of the object in the image.
- Two images with equal contrast or noise characteristics may still be perceived as being of different Quality. It is possible, for instance, to recognize an object with SNR < 1, i.e., where the noise exceeds the signal if the object is large enough.

Fig. 4.5 Four images have the same noise level, noise characteristics, and contrast. Object-dependent features such as the size of the object or the sharpness of boundaries still cause differences in the perceptibility of depicted objects

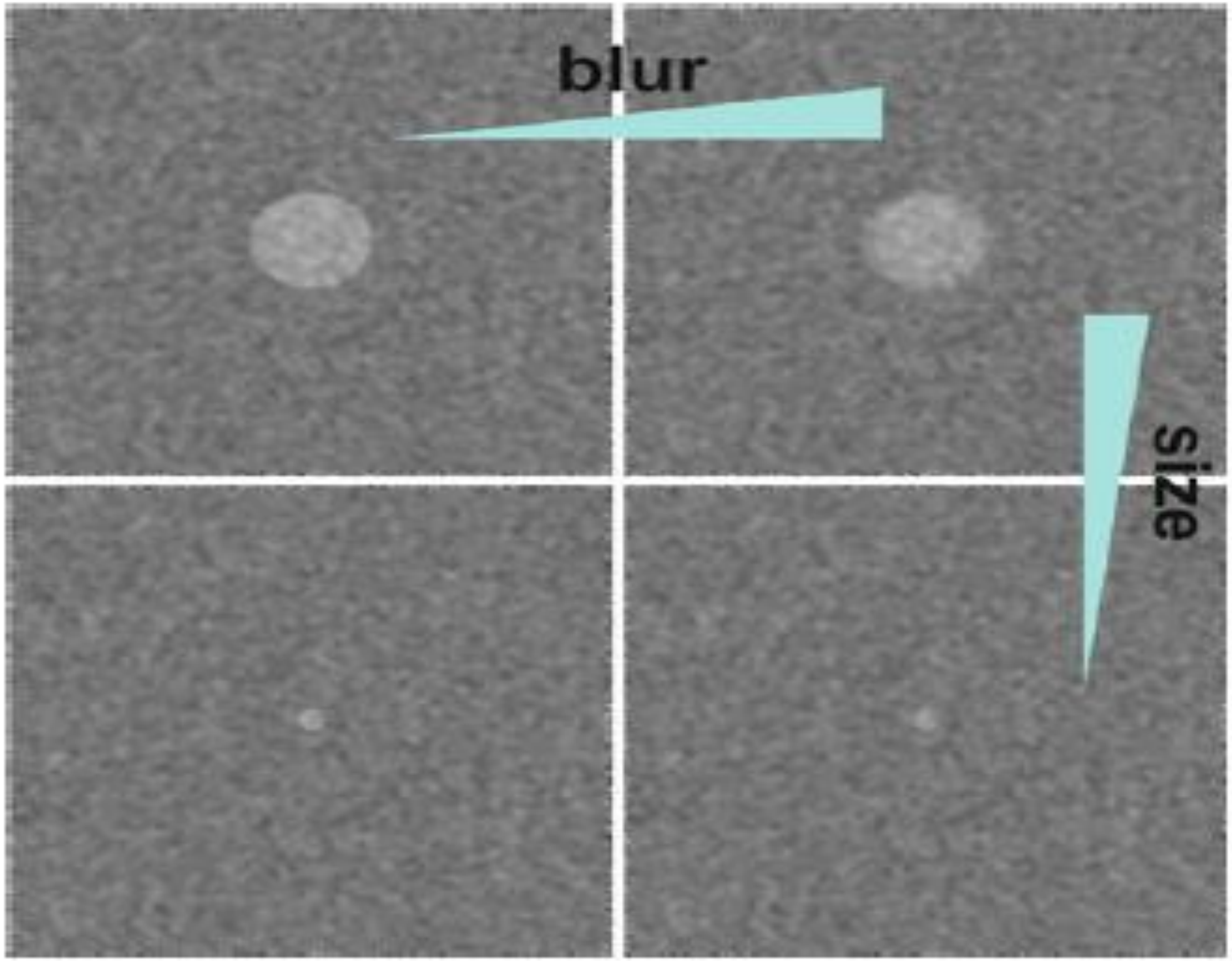


Image Enhancement in the Spatial Domain

= Image enhancement using processes performed in the Spatial domain resulting in images in the Spatial domain. We can written as

$$g(x, y) = T\left[f(x, y)\right]$$

where f(x,y) is an original image, g(x,y) is an output and T[ ] is a function defined in the area around (x,y)

Note: T[ ] may have one input as a pixel value at (x,y) only or multiple inputs as pixels in neighbors of (x,y) depending in each function.

Ex. Contrast enhancement uses a pixel value at (x,y) only for an input while smoothing filte use several pixels around (x,y) as inputs.

Types of Image Enhancement in the Spatial Domain

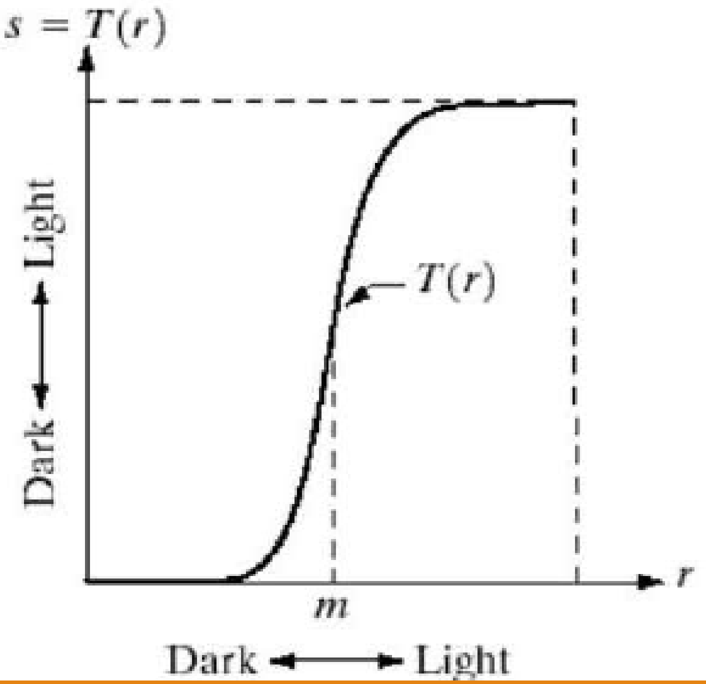
- Single pixel methods
  - Gray level transformations
    Example
    - Historgram equalization
    - Contrast stretching
  - Arithmetic/logic operations
    Examples
    - Image subtraction
    - Image averaging
- Multiple pixel methods
  Examples
  Spatial filtering
  - Smoothing filters
  - Sharpening filters

Gray Level Transformation

Transforms intensity of an original image into intensity of an output image using a function:

$$s = T(r)$$

where r = input intensity and s = output intensity



Example: Contrast enhancement

Image Enhancement Techniques

- **Contrast Enhancement:**
- The simplest method increases global contrast. If the range of possible intensity values Imin to Imax exceeds the range of intensities fmin to fmax in an image f, linear contrast enhancement is carried out creating new values g for every pixel by

$$g(f) = (f - f_{\min})\frac{I_{\max} - I_{\min}}{f_{\max} - f_{\min}} + I_{\min}.$$

- The function to map f on g is called transfer function. Contrast enhancement in an arbitrary intensity window wmin to wmax with Imin < wmin < wmax < Imax can be achieved with a similar transfer function. As there may be pixels with values f outside the window,

$$g(i,j) = \begin{cases} I_{\min}, & \text{if} \quad f(i,j) < w_{\min}, \\ (f(i,j) - w_{\min})\frac{I_{\max}}{w_{\max} - w_{\min}} + I_{\min}, & \text{if} \quad w_{\min} \leq f(i,j) \leq w_{\max}, \\ I_{\max}, & \text{if} \quad f(i,j) > w_{\max}. \end{cases}$$

- Output intensity of structures with input intensities outside of [wmin, wmax] is either Imin or Imax so that they are no longer recognizable.
- Contrast is improved for all other structures.
- **This kind of enhancement is routinely used for mapping a 16-bit-per-pixel intensity image (such as CT or MRI) onto an 8-bit range.**
- It is called windowing with window size wmax – wmin and level (wmax + wmin)/2.

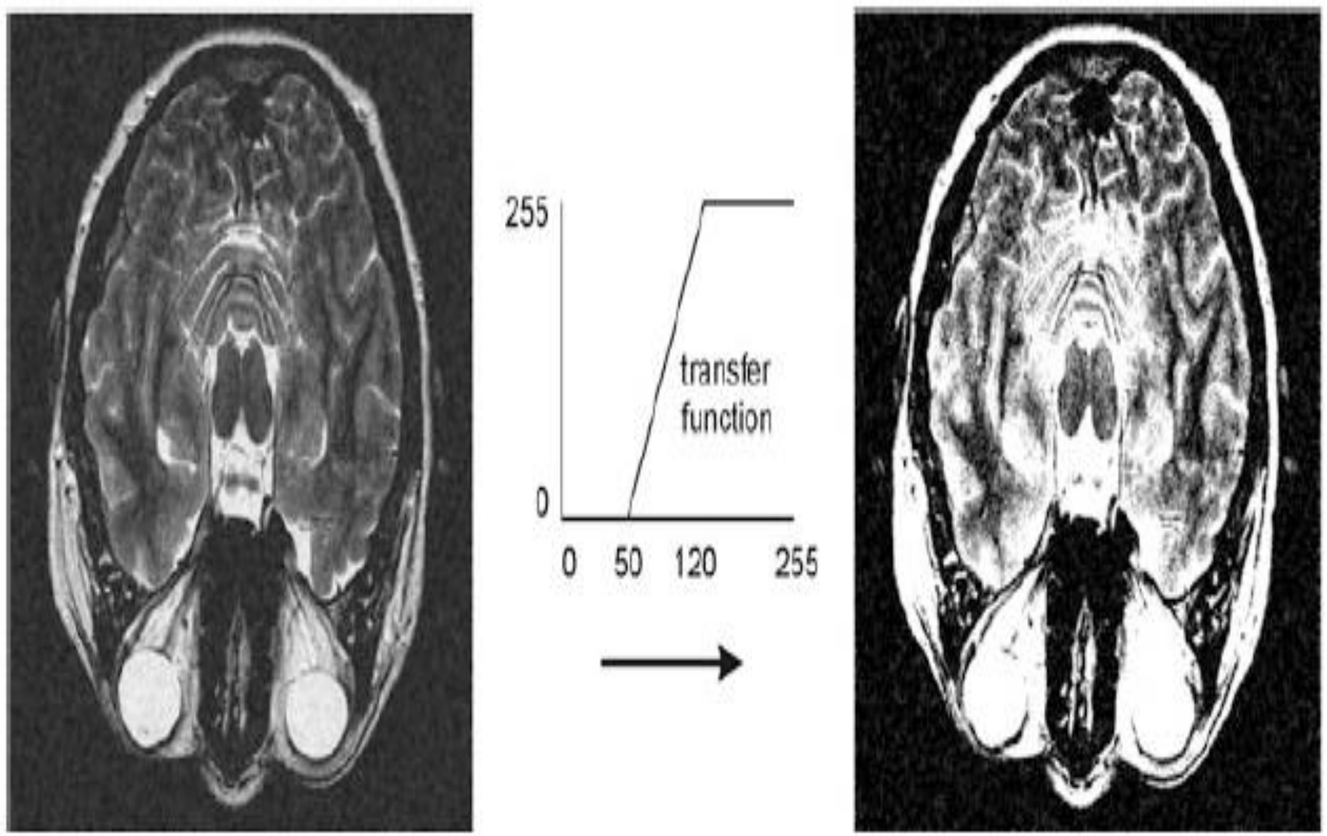


Fig. 4.6 Linear contrast enhancement in a window (50, 120) for enhancing soft tissue differences in an MR image. The enhancement comes at the cost of reducing contrast in regions outside of the window (such as the water in the eyeballs)

Contrast Stretching

- M-input image, l-output image. Ni is no.of pixels having ith gray level and frequency table is:

i	0	1	2	3	4	5	6	7
Ni	0	0	a	b	c	d	e	0

- Compute output image frequency table.
-

i	0	1	2	3	4	5	6	7
Ni - new	a	0	b	0	c	d	0	e

Histogram Processing

Let r_k, for $k = 0,1,2,\ldots,L-1$, denote the intensities of an L-level digital image, $f(x,y)$. The *unnormalized histogram* of f is defined as

$$h(r_k) = n_k \quad \text{for } k = 0,1,2,\ldots,L-1 \tag{3-6}$$

where n_k is the number of pixels in f with intensity r_k, and the subdivisions of the intensity scale are called *histogram bins*. Similarly, the *normalized histogram* of f is defined as

$$p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN} \tag{3-7}$$

For discrete values, we work with probabilities and summations instead of probability density functions and integrals (but the requirement of monotonicity stated earlier still applies). Recall that the probability of occurrence of intensity level r_k in a digital image is approximated by

$$p_r(r_k) = \frac{n_k}{MN} \tag{3-14}$$

where MN is the total number of pixels in the image, and n_k denotes the number of pixels that have intensity r_k. As noted in the beginning of this section, $p_r(r_k)$, with $r_k \in [0, L-1]$, is commonly referred to as a normalized image histogram.

The discrete form of the transformation in Eq. (3-11) is

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) \qquad k = 0, 1, 2, \ldots, L-1 \tag{3-15}$$

where, as before, L is the number of possible intensity levels in the image (e.g., 256 for an 8-bit image). Thus, a processed (output) image is obtained by using Eq. (3-15) to map each pixel in the input image with intensity r_k into a corresponding pixel with level s_k in the output image, This is called a *histogram equalization* or *histogram*

histogram equalization

- Suppose that a 3-bit image (L = 8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution in Table 3.1, where the intensity levels are integers in the range [0,L − 1] = [0, 7].
- Values of the histogram equalization transformation function are obtained using Eq. (3-15). For instance,

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

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

Similarly, $s_1 = T(r_1) = 3.08$, $s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, and $s_7 = 7.00$. This transformation function has the staircase shape shown in Fig. 3.19(b).

At this point, the s values are fractional because they were generated by summing probability values, so we round them to their nearest integer values in the range $[0, 7]$:

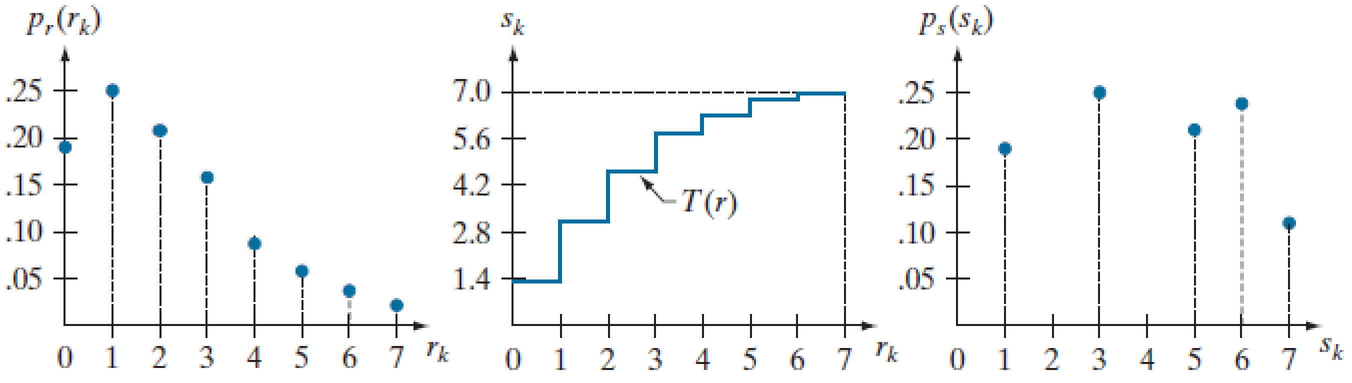
$$
\begin{array}{llll}
s_0 = 1.33 \rightarrow 1 & s_2 = 4.55 \rightarrow 5 & s_4 = 6.23 \rightarrow 6 & s_6 = 6.86 \rightarrow 7 \\
s_1 = 3.08 \rightarrow 3 & s_3 = 5.67 \rightarrow 6 & s_5 = 6.65 \rightarrow 7 & s_7 = 7.00 \rightarrow 7
\end{array}
$$

These are the values of the equalized histogram. Observe that the transformation yielded only five distinct intensity levels. Because $r_0 = 0$ was mapped to $s_0 = 1$, there are 790 pixels in the histogram equalized image with this value (see Table 3.1). Also, there are 1023 pixels with a value of $s_1 = 3$ and 850 pixels with a value of $s_2 = 5$. However, both r_3 and r_4 were mapped to the same value, 6, so there are $(656 + 329) = 985$ pixels in the equalized image with this value. Similarly, there are $(245 + 122 + 81) = 448$ pixels with a value of 7 in the histogram equalized image. Dividing these numbers by $MN = 4096$ yielded the equalized histogram in Fig. 3.19(c).

histogram equalization

a b c

FIGURE 3.19 Histogram equalization. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.



- Entropy is enhanced using histogram equalization.
- Assuming that intensities of some image f with histogram H(f) are defined on a continuous domain, histogram equalization maximizes entropy by creating a new image g with a constant histogram.
- The transfer function is

$$g(i,j) = \int_{I_{min}}^{g(i,j)} H_{norm}(g)\,dg.$$

- Intensities are quantized and, hence, histogram equalization is approximated by

$$g(i,j) = (I_{max} - I_{min} + 1) \sum_{g=I_{min}}^{g(i,j)} H_{norm}(g) - 1.$$

- The quantized version of histogram equalization of Eq. (4.16) can only decrease but not increase the entropy. Mapping of two or more intensity values on the same new intensity value will decrease the entropy.
- histogram equalization may increase the “perceived” information content by a human observer

histogram equalization

- Increase in entropy refers only to the perception of the image by a human observer.
- For the computer, the **decrease in entropy is definitely a loss of information.**
- For using histogram equalization as preprocessing in computer-based image analysis, the gain in local contrast should be balanced against this information loss.
- **Even for perception by a human observer, histogram equalization not always achieves the desired effect. It comes at the cost of reduced information for intensities that rarely occur.**
- Entropy does not consider any information about objects in an image and each pixel is equally important.
- For instance, the large and dark background region of the MR image in Fig. 4.8 is enhanced at the cost of reducing contrast in the foreground.

adaptive histogram equalization (AHE)

- Histogram equalization can be improved by making it locally adaptive.
- In adaptive histogram equalization (AHE), local histograms are computed separately for every pixel from the intensity distribution in some neighborhood around this pixel. The resulting mapping is carried out separately for each pixel.
- Adaptive histogram equalization will also lead to an overamplification of noise in low contrast regions.
- Contrast limited adaptive histogram equalization (CLAHE) solves this problem by clipping the histogram at some level and redistributing the clipped area equally to all bins of the histogram.
- The clipping reduces the increase in the cumulative histogram and thus the contrast enhancement in regions with low contrast.

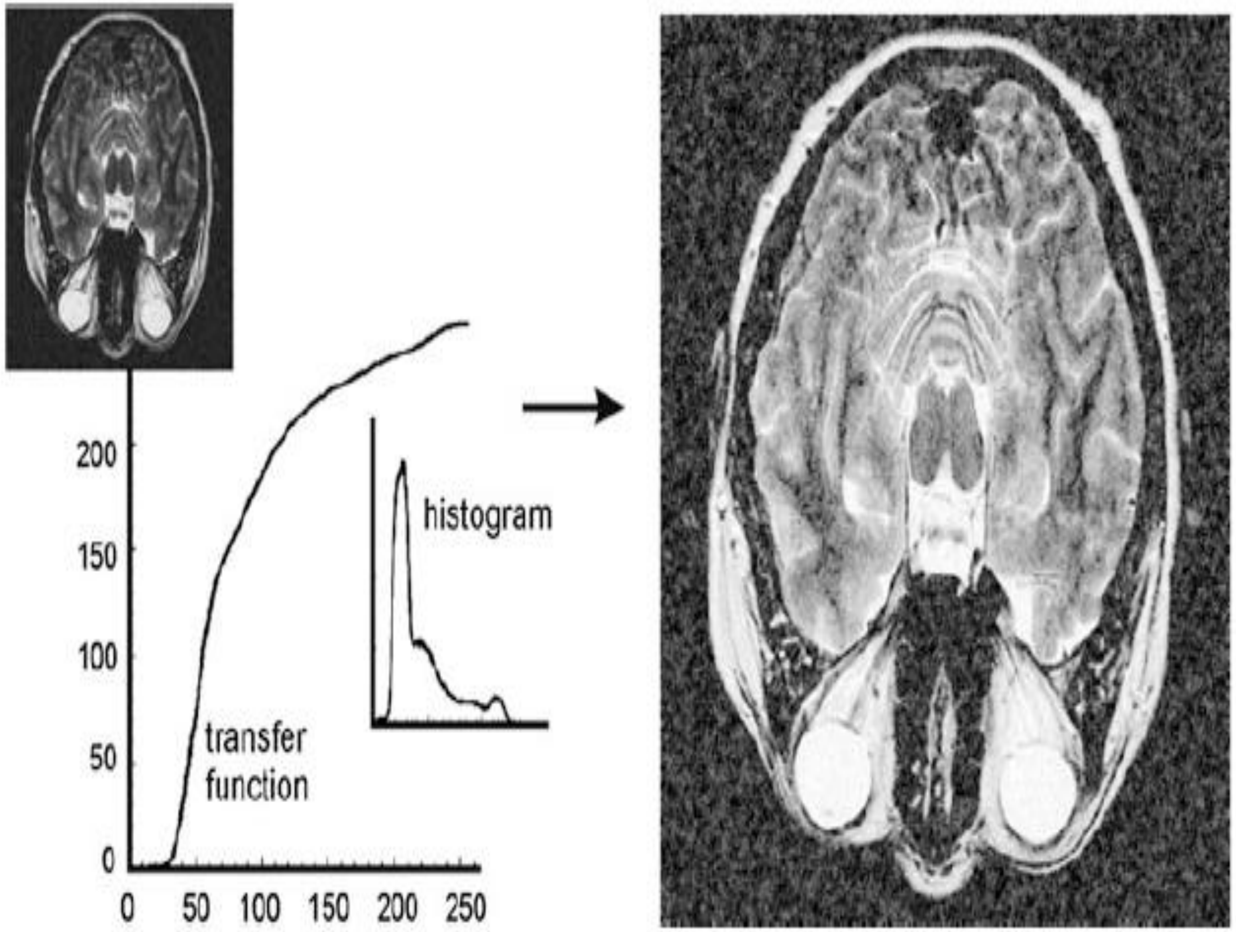
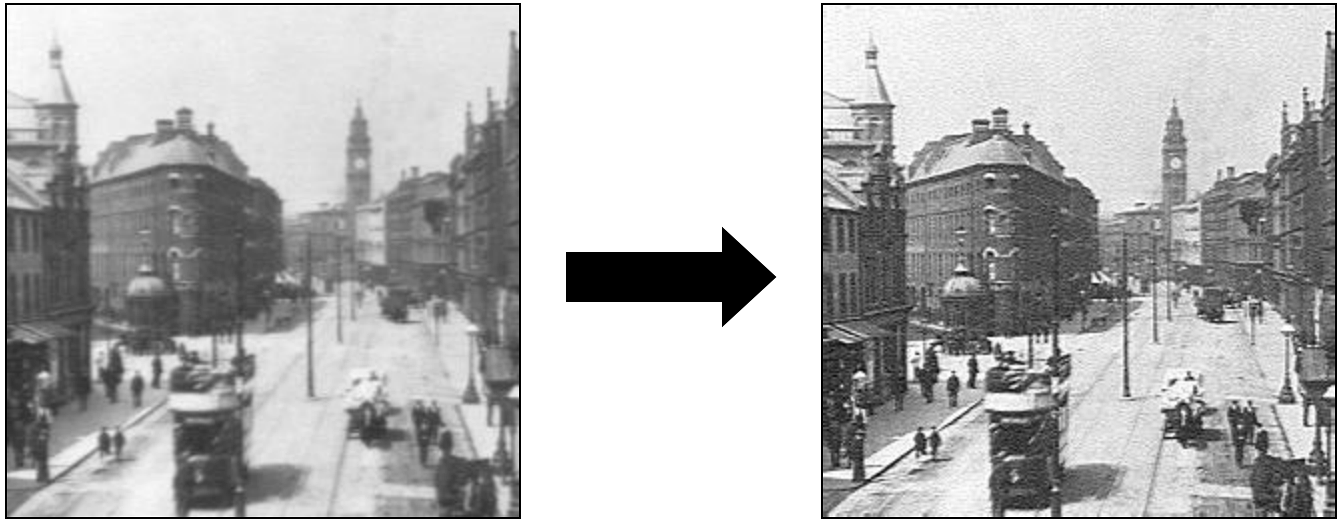


Fig. 4.8 Histogram equalization may produce unwanted effects since it enhances contrast on the assumption that often occurring gray values carry the most information. In the case depicted in this figure, it leads to the enhancement of background while reducing foreground contrast

What is Edge-enhancement?

- Physcophysical experiments indicate that an image with accentuated or crispened edges is often more subjectively pleasing than the original image.



Laplacian filter example

- Compute the convolution of the Laplacian kernels L_4 and L_8 with the image
- Use border values to extend the image

0	0	10	10	10
0	0	10	10	10
0	0	10	10	10
0	0	10	10	10
0	0	10	10	10

y

x

-1	-1	-1
-1	8	-1
-1	-1	-1

L_8

0	-1	0
-1	4	-1
0	-1	0

L_4

L_8

0	-30	30	0	0
0	-30	30	0	0
0	-30	30	0	0
0	-30	30	0	0
0	-30	30	0	0

L_4

0	-10	10	0	0
0	-10	10	0	0
0	-10	10	0	0
0	-10	10	0	0
0	-10	10	0	0

Laplacian filter example

- Compute the convolution of the Laplacian kernels L_4 and L_8 with the image
- Use zero-padding to extend the image

0	0	10	10	10
0	0	10	10	10
0	0	10	10	10
0	0	10	10	10
0	0	10	10	10

y

x

L_8

0	-20	50	50	50
0	-30	30	0	30
0	-30	30	0	30
0	-30	30	0	30
0	-20	50	50	50

L_4

0	-10	20	10	20
0	-10	10	0	10
0	-10	10	0	10
0	-10	10	0	10
0	-10	20	10	20

-1	-1	-1
-1	8	-1
-1	-1	-1

L_8

0	-1	0
-1	4	-1
0	-1	0

L_4

Resolution Enhancement

- In its simplest variant, **interpolation** is carried out as 1d linear or cubic interpolation in the direction of the z-axis. Interpolation is improved if structures to be interpolated are already segmented and the data is binary. Then, shape-based interpolation on the segment boundary can be carried out.
- **Image Interpolation:**
- Interpolation is used in tasks such as zooming, shrinking, rotating, and geometrically correcting digital images.
- ***Interpolation* is the process of using known data to estimate values at unknown locations.**
- Suppose that an image of size 500 * 500 pixels has to be enlarged 1.5 times to 750 * 750 pixels. A simple way to visualize zooming is to create an imaginary 750 * 750 grid with the same pixel spacing as the original image, then shrink it so that it exactly overlays the original image.
- the pixel spacing in the shrunken 750 * 750 grid will be less than the pixel spacing in the original image. To assign an intensity value to any point in the overlay, we look for its closest pixel in the underlying original image and assign the intensity of that pixel to the new pixel in the 750 * 750 grid.
- When intensities have been assigned to all the points in the overlay grid, we expand it back to the specified size to obtain the resized image.

Interpolation

- The method just discussed is called ***nearest neighbor interpolation*** because it assigns to each new location the intensity of its nearest neighbor in the original image
- This approach is simple but, it has the tendency to produce undesirable artifacts, such as severe distortion of straight edges.
- A more suitable approach is ***bilinear interpolation,*** in which we use the four nearest neighbors to estimate the intensity at a given location. Let (x, y) denote the coordinates of the location to which we want to assign an intensity value and let $v(x, y)$ denote that intensity value.
- For bilinear interpolation, the assigned value is obtained using the equation
- $$v(x, y) = ax + by + cxy + d$$
- where the four coefficients are determined from the four equations in four unknowns that can be written using the *four* nearest neighbors of point (x, y). Bilinear interpolation gives much better results than nearest neighbor interpolation, with a modest increase in computational burden.

shape-based interpolation

- Approaches exist to apply shape-based interpolation to gray level images by inferring a shape gradient between slices.
- The shape gradient is defined as the direction between slices in which the smallest gray level change occurs. This can be computed for each voxel and defines a displacement field between two slices.
- If new slices are to be interpolated, three steps are carried out:
- • Estimate displacement vectors for each voxel in the interpolated slice.
- • Compute corresponding voxel in the two slices above and below the interpolated slice.
- • Linearly interpolate voxel values in the interpolated slice along the interpolation direction.
- Features in the low-resolution image are transferred to the high resolution realm using an interpolated high-resolution image as spatial reference

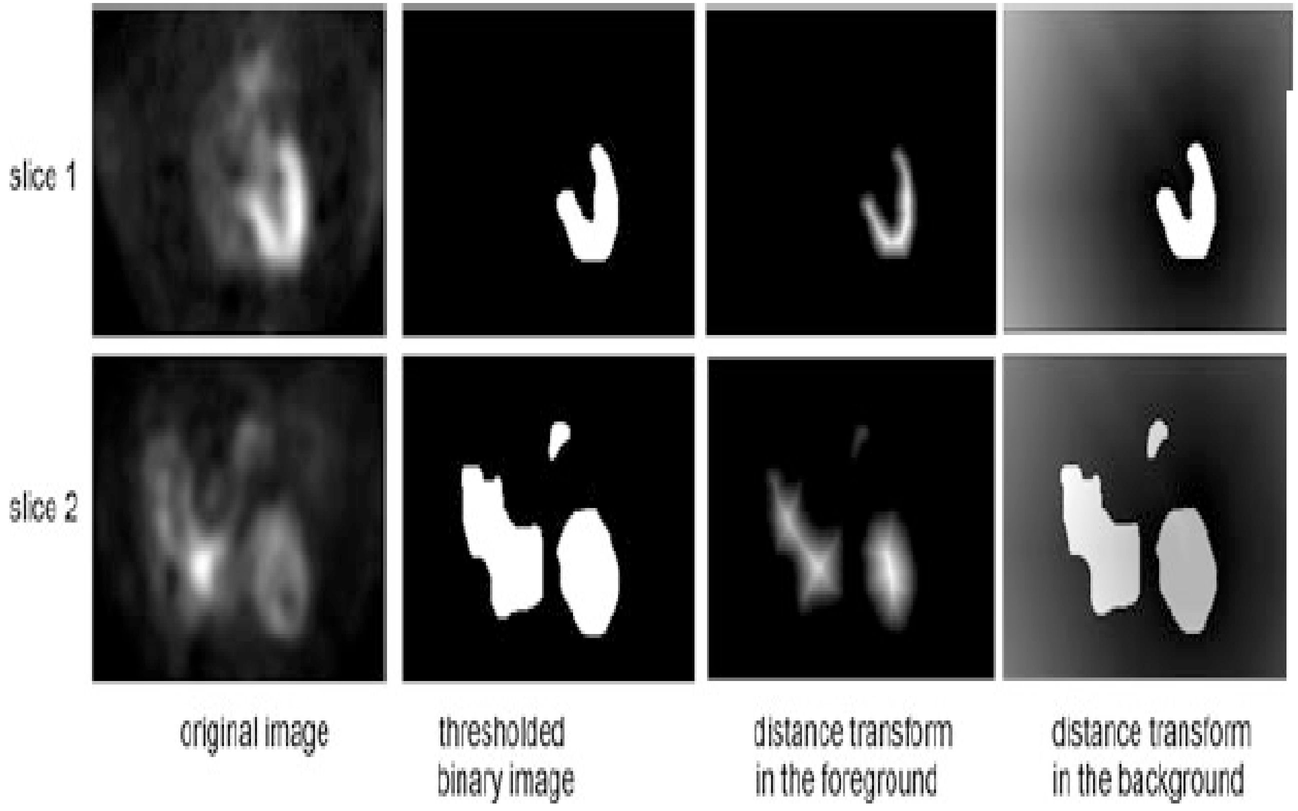


Fig. 4.9 Preparations for shape-based interpolation between two binary images: Computation of the binary image from the original (in this case SPECT images from the left ventricle of the heart), computation of a distance transform in the foreground (positive distance values) and in the background (negative distance values)

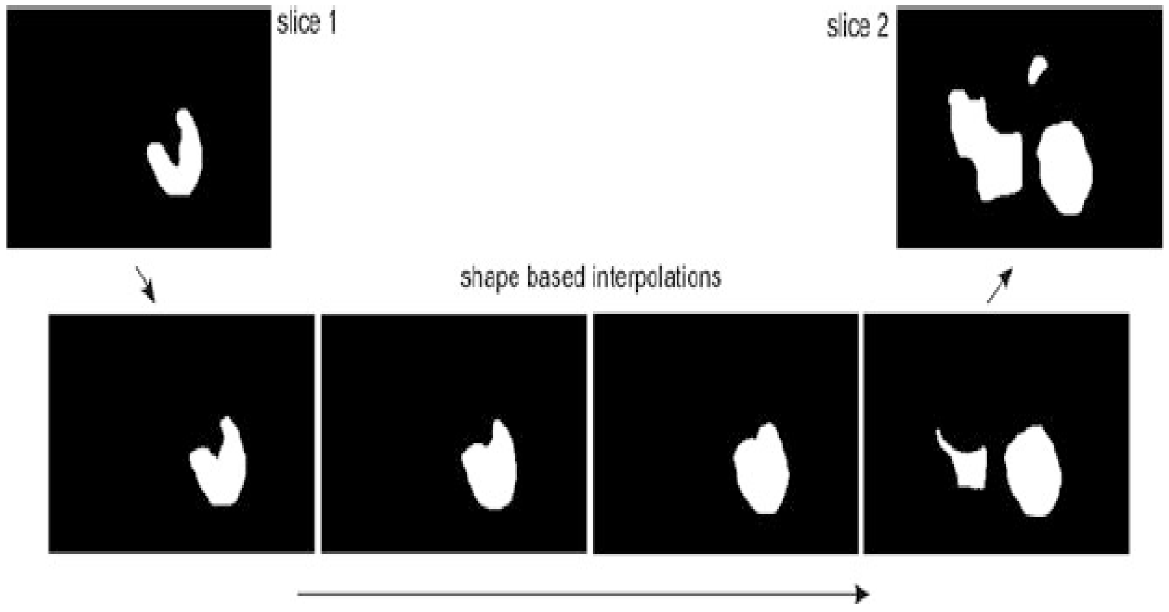


Fig. 4.10 Result of interpolated intermediate slices from the binary images in Fig. 4.9

Sobel filter

- The Sobel filter is used for edge detection.
- It works by calculating the gradient of image intensity at each pixel within the image. It finds the direction of the largest increase from light to dark and the rate of change in that direction.
- The result shows how abruptly or smoothly the image changes at each pixel, and therefore how likely it is that that pixel represents an edge.
- It also shows how that edge is likely to be oriented. The result of applying the filter to a pixel in a region of constant intensity is a zero vector.
- The result of applying it to a pixel on an edge is a vector that points across the edge from darker to brighter values.

The sobel filter uses two 3 x 3 kernels. One for changes in the horizontal direction, and one for changes in the vertical direction.

The two kernels are convolved with the original image to calculate the approximations of the derivatives.

If we define Gx and Gy as two images that contain the horizontal and vertical derivative approximations respectively, the computations are:

$$G_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * A \quad \text{and} \quad G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} * A$$

Where A is the original source image.

The x coordinate is defined as increasing in the right-direction and the y coordinate is defined as increasing in the down-direction.

Sobel Operator

At each pixel in the image, the gradient approximations given by G_x and G_y are combined to give the gradient magnitude, using:

$$G = \sqrt{G_x^2 + G_y^2}$$

The gradient's direction is calculated using:

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

A Θ value of 0 would indicate a vertical edge that is darker on the left side.

Sobel filter example

- Compute Gx and Gy, gradients of the image performing the convolution of Sobel kernels with the image
- Use zero-padding to extend the image

y

0	0	10	10	10
0	0	10	10	10
0	0	10	10	10
0	0	10	10	10
0	0	10	10	10

x

G_x

0	30	30	0	-30
0	40	40	0	-40
0	40	40	0	-40
0	40	40	0	-40
0	30	30	0	-30

G_y

-10	-30	-40	-30	-10
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
10	30	40	30	10

1	0	-1
2	0	-2
1	0	-1

h_x

-1	-2	-1
0	0	0
1	2	1

h_y

Sobel filter example

- Compute Gx and Gy, gradients of the image performing the convolution of Sobel kernels with the image
- Use border values to extend the image

0	0	10	10	10
0	0	10	10	10
0	0	10	10	10
0	0	10	10	10
0	0	10	10	10

y

x

G_x

0	40	40	0	0
0	40	40	0	0
0	40	40	0	0
0	40	40	0	0
0	40	40	0	0

G_y

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

1	0	-1
2	0	-2
1	0	-1

h_x

-1	-2	-1
0	0	0
1	2	1

h_y

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

	0	0		
	0	0		
	0	0		
	0	0		
	0	0		

Laplacian Masks

Used for estimating image Laplacian $\nabla^2 P = \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2}$

-1	-1	-1
-1	8	-1
-1	-1	-1

0	-1	0
-1	4	-1
0	-1	0

→ The center of the mask is positive

or

1	1	1
1	-8	1
1	1	1

0	1	0
1	-4	1
0	1	0

→ The center of the mask is negative

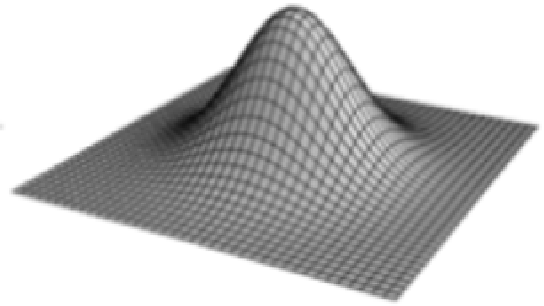
Application: Enhance edge, line, point

Disadvantage: Enhance noise

Image Enhancement in Spatial Domain

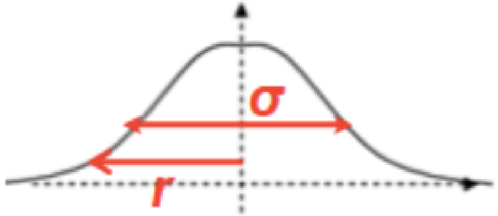
Spatial Filtering

$$G_\sigma(r) = e^{-\frac{r^2}{2\sigma^2}} \quad \text{or} \quad G_\sigma(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$



- where
  - $\boldsymbol{\sigma}$ is width (standard deviation)
  - $\boldsymbol{r}$ is distance from center

Gaussian Filter



0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

Gaussian filter

Gaussian kernel

or the use of a Gaussian kernel. The partial derivatives of the Gaussian with standard deviation σ are

$$D_{\text{Gauss},x} = \frac{\partial f_{\text{Gauss}}}{\partial x}(x,y) = -\frac{x}{\sigma^3\sqrt{2\pi}}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4.23}$$

and

$$D_{\text{Gauss},y} = \frac{\partial f_{\text{Gauss}}}{\partial y}(x,y) = -\frac{y}{\sigma^3\sqrt{2\pi}}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{4.24}$$

Gaussian Kernal

Let's use a 3x3 Gaussian kernel as an example:

Gaussian Kernel:

1/16 1/8 1/16

1/8 1/4 1/8

1/16 1/8 1/16

Image array=
[[20, 30, 40, 50, 60],
[25, 35, 45, 55, 65],
[30, 40, 50, 60, 70],
[35, 45, 55, 65, 75],
[40, 50, 60, 70, 80]]

```
source = cv2.GaussianBlur(f1, (3, 3), 0)
source
```

```
array([[28, 33, 43, 53, 58],
       [30, 35, 45, 55, 60],
       [35, 40, 50, 60, 65],
       [40, 45, 55, 65, 70],
       [43, 48, 58, 68, 73]], dtype=uint8)
```

Laplacian of Gaussian (LoG) filter

- The Laplacian is often used in conjunction with smoothing kernel because it is very sensitive to noise.
- A well-known kernel is the Laplacian of Gaussian (LoG) filter, which computes the Laplacian of a Gaussian function and then uses the result to convolve the image .
- The filter is sometimes called **Mexican hat filter** because of its shape.
- It is also known as Marr- Hildreth filter after David Marr and Ellen Hildreth, who showed that early edge enhancement in human vision can be modeled by an LoG filter.

$$\mathrm{LoG}(x,y) = -\frac{1}{\pi\sigma^4}\left(1-\frac{x^2+y^2}{2\sigma^2}\right)\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \qquad (4.27)$$

Laplacian of Gaussian (LoG) filter

- The Laplacian of Gaussian (LoG) filter is a popular edge detection and enhancement technique.
- It involves convolving the image with the Laplacian of a Gaussian kernel.
- **The Gaussian kernel is applied to smooth the image, reducing noise, while the Laplacian operator highlights regions of rapid intensity change, which correspond to edges.**

Consider the following grayscale image matrix:

20 30 40 50 60

25 35 45 55 65

30 40 50 60 70

35 45 55 65 75

40 50 60 70 80

```
dest = cv2.Laplacian(source, cv2.CV_32F, ksize=3)
dest
```

```
array([[ 56.,  36.,  16.,  -4., -24.],
       [ 52.,  32.,  12.,  -8., -28.],
       [ 40.,  20.,   0., -20., -40.],
       [ 32.,  12.,  -8., -28., -48.],
       [ 16.,  -4., -24., -44., -64.]], dtype=float32)
```

5 × 5 Laplacian of Gaussian mask

0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

cv2.getGaborKernel(ksize, sigma, theta, **lambda**, gamma, psi, ktype)

Gabor Filter

When a Gabor filter is applied to an image, it gives the highest response at edges and at points where texture changes. A Gabor filter responds to edges and texture changes.

cv2.getGaborKernel(ksize, sigma, theta, lambda, gamma, psi, ktype)

sigma is the standard deviation of the Gaussian function used in the Gabor filter. **theta** is the orientation of the normal to the parallel stripes of the Gabor function. **lambda** is the wavelength of the sinusoidal factor in the above equation. **gamma** is the spatial aspect ratio. **psi** is the phase offset.

ktype indicates the type and range of values that each pixel in the Gabor kernel can hold

Gábor filter

A filter that has been shown to be a model for directionally sensitive cells in the primary visual cortex is the Gábor filter.

$$G_{\sigma,\alpha,\gamma,\lambda,\psi}(x, y) = \exp\left(-\frac{s^2 + \gamma t^2}{2\sigma^2}\right) \cos\left(2\pi \frac{s}{\lambda} + \psi\right), \tag{4.29}$$

with

$$s = x \sin \alpha + y \cos \alpha, \quad t = x \cos \alpha - y \sin \alpha. \tag{4.30}$$

The exponential term is a smoothing function which is wider along s (i.e., in a direction α with respect to x) than along t perpendicular to s. The parameter

γ determines the elongatedness of the function. The cosine term produces the difference of values along s. Its wavelength is given by λ. It controls the width of the range along which the difference is taken. The parameter ψ is an offset for the difference function.

Filter banks of Gábor filters can be used to enhance and group edges by direction, curvedness, and steepness.

Types of Noise

Types of Noise

There are different types of noise in image processing. The common types of noises are:

1. Gaussian Noise:

Gaussian noise is one of the types of noise which follows the Gaussian distribution. It is defined by its mean and standard deviation values. Gaussian noise is usually caused by sensor noise or electronic interference.

2. Salt and Pepper Noise:

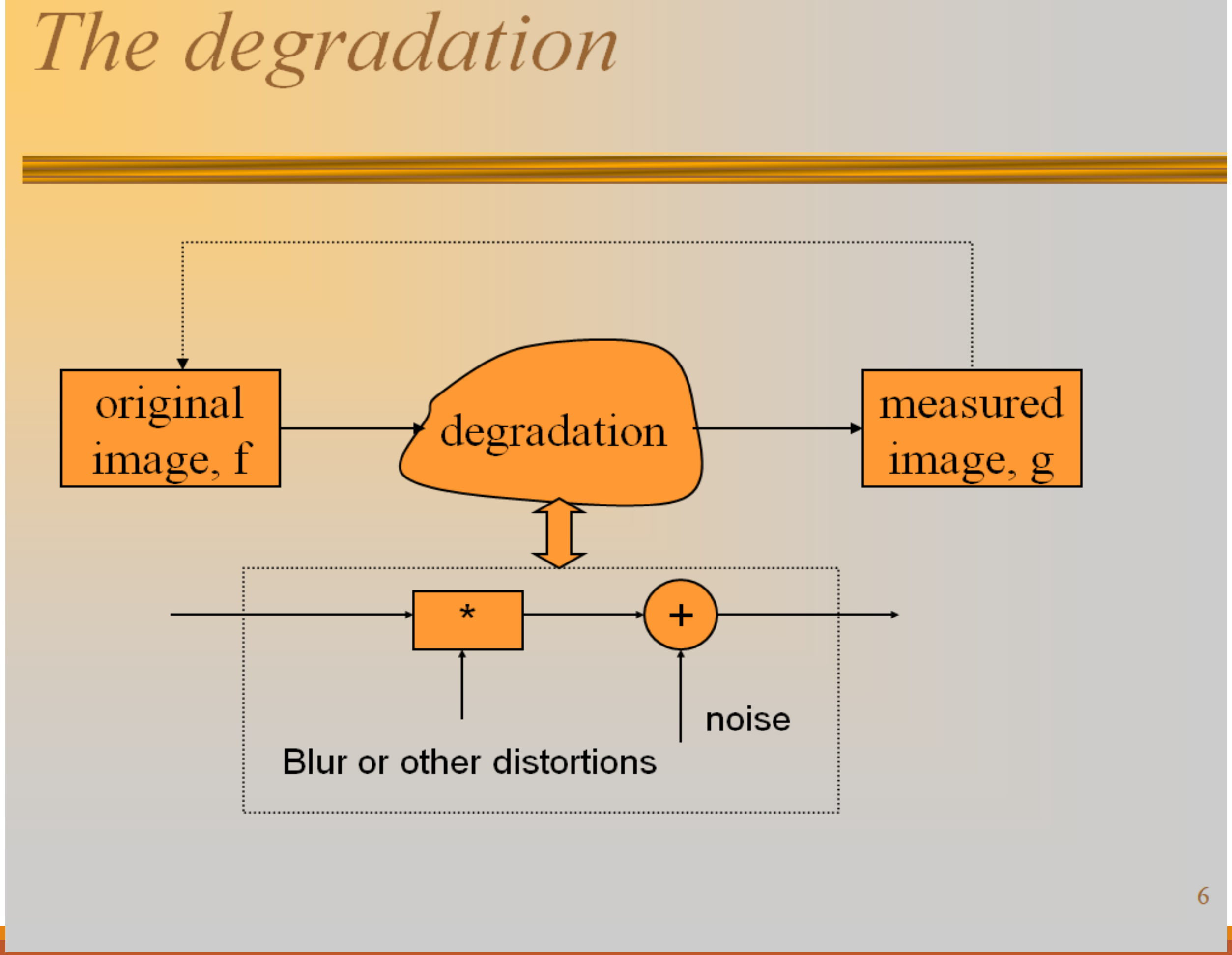
Salt and **pepper noise** is a type of noise that occurs as white and black pixels in the image. It is caused by errors in image acquisition or transmission.

3. Speckle Noise:

Speckle noise is a type of noise that occurs in images acquired by ultrasound or laser imaging systems. It is caused by the interference of the sound or light waves.

Noise Reduction

- Noise is usually modeled stationary, additive, and with zero mean.
- A noisy image g is related to the unknown noise-free image f through g = f + n, where n is the zero mean noise.
- Noise removal through linear filtering consists of estimating the expected value E(g).



Noise

Filtering is useful for noise reduction...

Common types of noise:

- **Salt and pepper noise**: random occurrences of black and white pixels
- **Impulse noise:** random occurrences of white pixels
- **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution

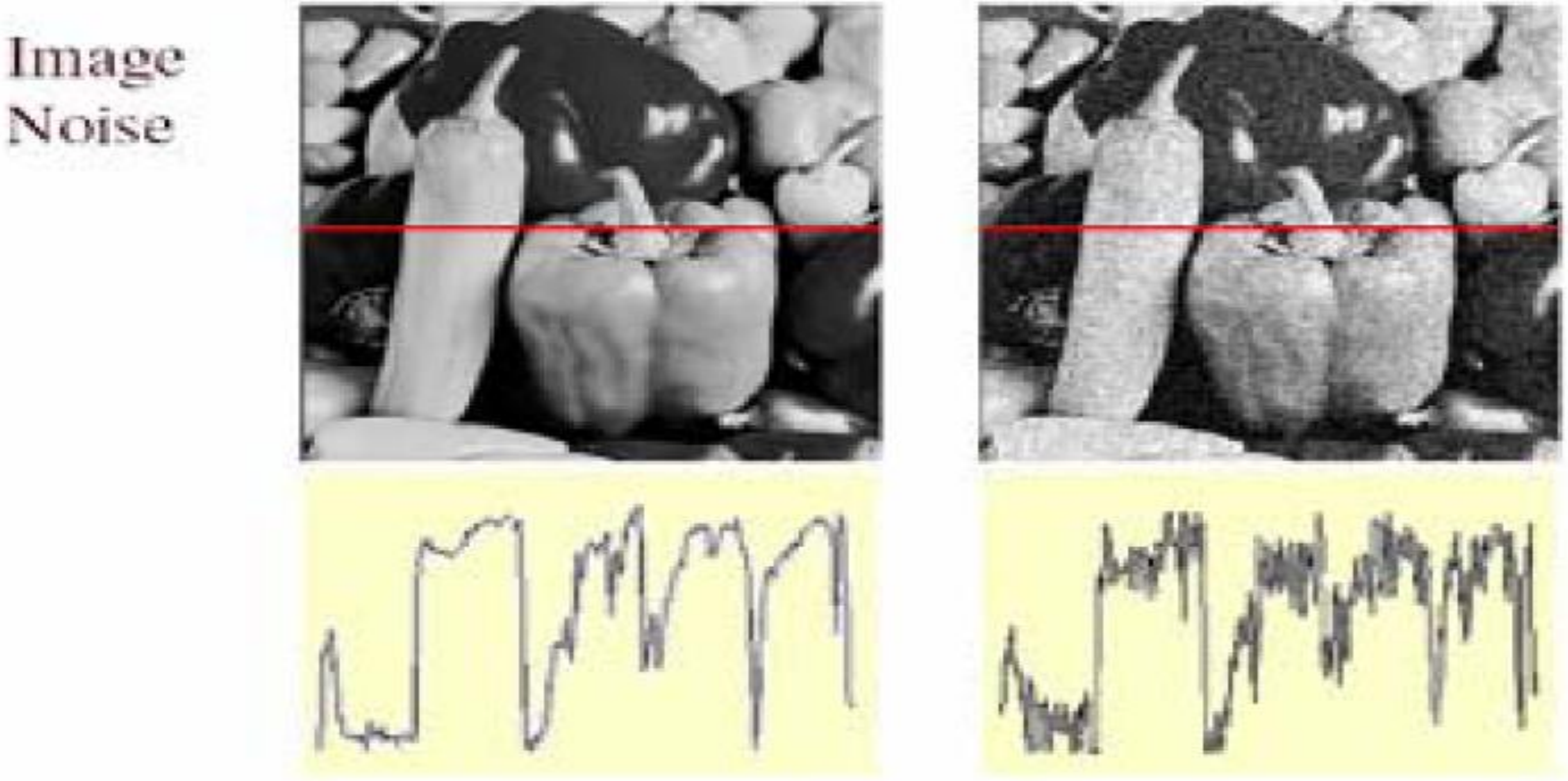
Original

Salt and pepper noise

Impulse noise

Gaussian noise

Gaussian noise



$$f(x,y) = \overbrace{\bar{f}(x,y)}^{\text{Ideal Image}} + \overbrace{\eta(x,y)}^{\text{Noise process}}$$

Gaussian i.i.d. ("white") noise:
$\eta(x,y) \sim \mathcal{N}(\mu,\sigma)$

noise reduction schemes

- noise reduction schemes try to reconstruct E(g) from various assumptions about f :
- • Linear filtering assumes f to be locally constant. E is then estimated by averaging over a local neighborhood.
- • Median filtering assumes that noise is normally distributed, f is locally constant, except for edges, the signal at edges is higher than the noise, and edges are locally straight.
- • Diffusion filtering and their approximations assume that f is locally constant, except at edges, and the properties of edges and noise can be differentiated by amplitude or frequency.
- • Bayesian image restoration requires f to be locally smooth, except for edges. It further requires that in some local neighborhood edge, pixels are not the majority of all pixels in that neighborhood.

Since most of the assumptions are not true everywhere in the image, filtering results in various filter-specific artefacts.

If f is constant in some neighborhood around (i, j), $E(i, j)$ can be estimated by averaging over this neighborhood. The operation can be carried out in the spatial domain as a convolution with a *mean filter* (sometimes called *boxcar filter*) of size s:

$$f(i,j) \approx \left[g * c_{\mathrm{mean},s}\right](i,j), \tag{4.32}$$

where "*" stands for the convolution operation. The convolution kernel $c_{\mathrm{mean},s}$ is a square matrix of size $s \times s$ with s being odd and the filter being centered:

$$c_{\mathrm{mean},s} = \frac{1}{s^2}\begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & & \dots & 1 \\ \dots & & & \dots \\ & & 1 & 1 \end{pmatrix}. \tag{4.33}$$

Mean filtering

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x, y]$

$G[x, y]$

Mean filtering

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x, y]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$G[x, y]$

Binomial Filter

Several other linear filters also deliver an estimate of $E(g)$. The *binomial filter* is a convolution kernel $c_{\mathrm{binom},b}$ whose one-dimensional version contains the binomial coefficients of order b. A 2d kernel can be computed from convolving two 1d filters. A 1d filter of order b is created from repetitive convolution of a 1d filter of order 1:

$$c^{1D}_{\mathrm{binom},b} = \frac{1}{2^b} c^{1D}_{\mathrm{binom},b-1} * c^{1D}_{\mathrm{binom},1}, \quad \text{with} \quad c^{1D}_{\mathrm{binom},1} = [\,1 \quad 1\,]. \tag{4.34}$$

The 2d filter $c_{\mathrm{binom},b}$ is computed by $c_{\mathrm{binom},b} = c^{1D}_{\mathrm{binom},1} \times \left(c^{1D}_{\mathrm{binom},1}\right)^{\mathrm{T}}$, where

T denotes the transpose of a matrix. The filter coefficients of an order i- binomial filter are the coefficients found in the i-th row of Pascal's triangle. The first few unnormalized filters b^i are listed in the following (superscript number denotes filter order):

- b^0=[1]
- b^1=[1,1]
- b^2=[1,2,1]
- b^3 = [1,3,3,1]

Other Linear Filters

- Filtering may be carried out by multiplication in the frequency domain.
- Noise in the frequency domain is modeled as a random process whose expected value for the amplitude is either constant (so-called white noise) or much slower decreasing with frequency than the amplitude of the signal (colored noise).
- In either case, noise dominates the high frequency range and noise reduction consists of applying a low-pass filter.
- The ideal low-pass filter with cut-off frequency wmax, however, produces severe ringing artefacts.
- **Examples for filters that do not cause ringing are the Gaussian, the Butterworth filter, the Hamming or the Hann window.**

The Gaussian for filtering in frequency space corresponds to a Gaussian in the spatial domain with inverted standard deviation. The *Butterworth filter* attenuates noise proportional to frequency with cut-off frequency $w_{\max}$:

$$C_{\text{Butterworth},\omega_{\max}}(u, v) = \frac{1}{1 + \left((u^2 + v^2)/w_{\max}^2\right)^k}. \tag{4.36}$$

The Hamming and Hann window with cut-off frequency $w_{\max}$ are two similar filter functions. They are defined by

$$C_{\text{Ha},\omega_{\max}}(u, v) = \begin{cases} \alpha - (1 - \alpha)\cos\left(\frac{\sqrt{u^2 + v^2}}{w_{\max}}\pi\right), & \text{if} \quad u^2 + v^2 < w_{\max}^2 \\ 0, & \text{otherwise} \end{cases} \tag{4.37}$$

The filter is called *Hamming* window if $\alpha = 0.53836$. For $\alpha = 0.5$ it is called *Hann* window.

Noise reduction in medical images

- Noise reduction in medical images may be difficult because often spatial resolution must not be reduced. Small structures or structures with a very convoluted surface must still be visible after noise removal.
- If, e.g., voxels in an MR brain image have an edge length of 1.0 mm, it may be expected that small gyrii are recognizable which are only few millimeters apart.
- Smoothing either requires a sufficiently high object-dependent contrast in the input image, a low level of spatial detail, or a combination of both.
- High contrast images are for instance MR soft tissue images, CT bone images, contrast-enhanced CT or MR angiograms, or some nuclear images.
- CT soft tissue images are difficult to smooth because of the low contrast between tissues. The same is true for ultrasound images because of the high ratio of low frequency noise and artefacts. Noise removal in nuclear images, such as SPECT, may also be inappropriate, since —despite of a good contrast—it may further reduce the already low spatial resolution.
- Linear filtering will generally produce poor results if the SNR is low or if the ratio of low frequency noise is high. Successful noise reduction in such cases requires an edge model as integral part of the smoothing process. Such methods are called edge-preserving smoothing.

ORDER-STATISTIC FILTERS

- order-statistic filters are spatial filters whose response is based on ordering (ranking) the values of the pixels contained in the neighborhood- ranking result determines the response of the filter.
- **Median Filter**
- The best-known order-statistic filter in image processing is the *median filter*, which, replaces the value of a pixel by the median of the intensity levels in a predefined neighborhood of that pixel:

$$\hat{f}(x,y) = \underset{(r,c)\in S_{xy}}{\text{median}}\{g(r,c)\}$$

- where, as before, *Sxy* is a subimage (neighborhood) centered on point (*x*, *y*). The value of the pixel at (*x*, *y*) is included in the computation of the median.
- This filter provides excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise.

Max and Min Filters

- The median represents the 50th percentile of a ranked set of numbers, while using the 100th percentile results in the so-called ***max filter,*** given by

$$\hat{f}(x,y) = \max_{(r,c)\in S_{xy}} \{g(r,c)\}$$

- This filter is useful for finding the brightest points in an image or for eroding dark regions adjacent to bright areas. Also, because pepper noise has very low values, it is reduced by this filter as a result of the max selection process in the subimage area *Sxy*.
- The **0th percentile filter is the *min filter*:**

$$\hat{f}(x,y) = \min_{(r,c)\in S_{xy}} \{g(r,c)\}$$

- This filter is useful for finding the darkest points in an image or for eroding light regions adjacent to dark areas. Also, it reduces salt noise as a result of the min operation.

Filtering Process- sample exercise

0	1	0	6	7
2	0	1	6	5
1	1	**7**	5	6
1	0	6	6	5
2	5	6	7	6

- Mean filter= 3.55 ==4
- Median filter = 5
- Min filter ==0
- Max filter = 7.

Diffusion Filtering

- **The median filter has two disadvantages:** It does not remove noise at edges even if the edge follows the implicit edge model, and it may alter edges in a random fashion that does not follow the edge model.
- **Diffusion filtering is an alternative which enables smoothing at edges.** It may also accommodate a wide range of edge models.
- Diffusion filtering uses diffusion of a liquid or gaseous material as a model for noise reduction.
- For applying a diffusion process as filter, image intensity is taken as material density. Noise is taken as density variation and diffusion is carried out iteratively.
- After an infinite number of iterations, **homogeneous diffusion** levels any density inhomogeneity resulting in a noise-free image without edges. It can be shown that the result from iterated homogeneous diffusion can be computed by filtering the image with a Gaussian with a standard deviation that depends on the (artificial) time passed for the diffusion process. Hence, homogeneous diffusion does not require iterative computation.

Diffusion Filtering

- Diffusion across edges should be inhibited for edge enhancement. Since boundaries are unknown (otherwise, edge-preserving smoothing would be trivial), the edge response from edge enhancement is used to indicate potential boundary locations.
- **Inhomogeneous diffusion treats such boundary locations as a semi-permeable material.**
- Unlike homogeneous diffusion, computation of the diffusion result is not possible in a single step but requires iterative computation along the time axis.
- Preserving boundaries may be improved if, instead of restricting any kind of diffusion at edges, it is only restricted across edges. This is called **anisotropic diffusion.**
- Allowing diffusion parallel to an edge enables noise removal by smoothing while inhibiting diffusion across an edge. Gradient direction is used as discriminative feature between noise and edges.

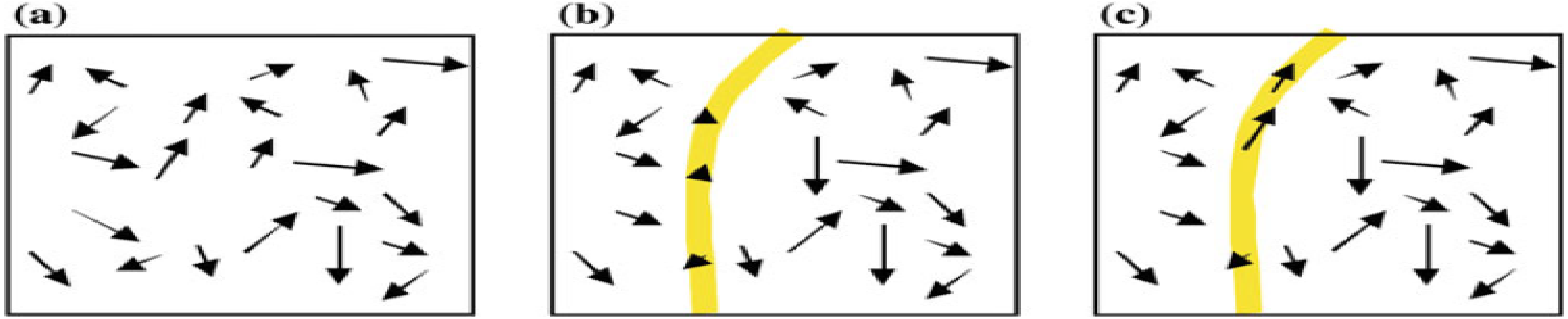


Fig. 4.19 **a** Homogeneous diffusion is only dependent on the density gradient, **b** inhomogeneous diffusion decreases at edges, **c** anisotropic diffusion decreases at an edge in edge normal direction

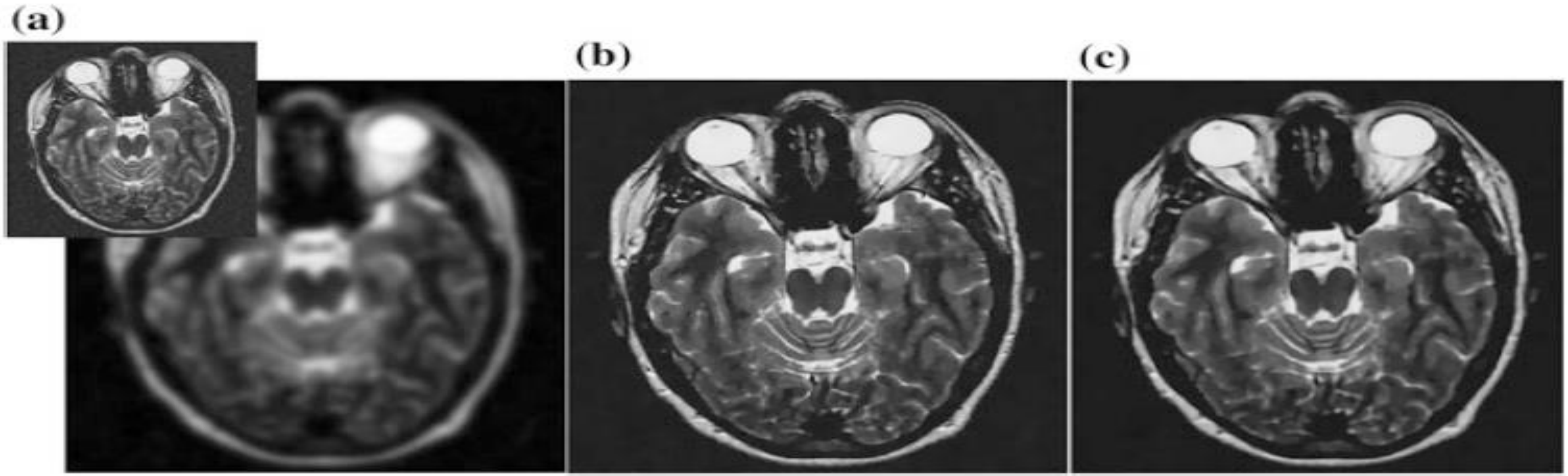


Fig. 4.20 Comparison of the different types of diffusion: **a** homogeneous diffusion, **b** inhomogeneous diffusion, **c** anisotropic inhomogeneous diffusion

Gradient Adaptive Smoothing

- Gradient adaptive smoothing is an iterative process that bears similarities to diffusion filtering.
- At each iteration, a new image is generated by weighted filtering with weights depending on the local gradient.
- **Bilateral Filtering:**
- Anisotropic diffusion filtering and adaptive smoothing are iterative procedures.
- Bilateral filtering attempts to approximate this by a single-step procedure.
- Since anisotropic diffusion restricts diffusion across potential segment boundaries, Gaussian smoothing is adapted such that the convolution kernel is additionally weighted by the intensity difference between central pixel and pixels in a predefined neighborhood