

Chapter

1

Introduction to Digital Image Processing

1.1 Digital Images

Visible light is essentially electromagnetic radiation with wavelengths between 400 and 700 nm. Each wavelength corresponds to a different color. On the other hand, a particular color does not necessarily correspond to a single wavelength. Purple light, for example, is a combination of red and blue light. In general, a color is characterized by a spectrum of different wavelengths.

The human retina contains three types of photoreceptor cone cells, which transform the incident light with different color filters. Because there are three types of cone receptors, three numbers are necessary and sufficient to describe any perceptible color. Hence, it is possible to produce an arbitrary color by superimposing appropriate amounts of three primary colors, each with its specific spectral curve. In an additive color reproduction system, such as a color monitor, these three primaries are red, green, and blue light. The color is then specified by the amounts of red, green, and blue (RGB). Equal amounts of red, green, and blue yield white (see Figure 1.1(a)). Ideal white light has a flat spectrum in which all wavelengths are present. In practice, white light sources approximate this property. In a subtractive color reproduction system, such as printing or painting, these three primaries typically are cyan, magenta, and yellow (CMY). Cyan is the color of a material, seen in white light, that absorbs red and reflects green and blue, and can thus be obtained by additive mixing of equal amounts of green and blue light. Similarly, magenta is the result of the absorption of green light and consists of equal amounts of red and blue light, and yellow is the result of the absorption of blue and consists of equal amounts of red and green light. Therefore, painting a white wall with a mixture of cyan and magenta yields a blue color; using cyan and yellow gives green, and yellow with magenta gives red. Mixing yellow, cyan, and magenta paint produces black (i.e., only absorption and no reflection)

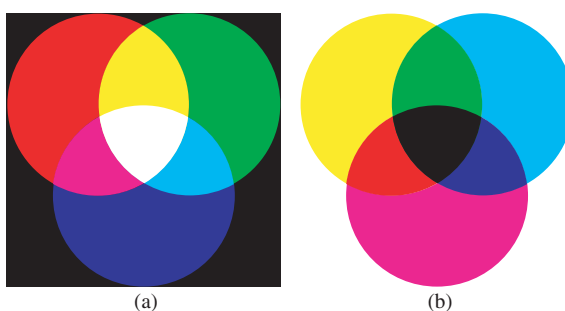


Figure 1.1 Color mixing: (a) additive color mixing (b) subtractive color mixing.

(see Figure 1.1(b)). Nevertheless, in print applications black ink is typically added as a separate fourth basic color for a variety of practical and quality reasons. Black is abbreviated to “K,” yielding the CMYK color model.

Note that equal distances in physical intensity are not perceived as equal distances in *brightness*. Intensity levels should rather be spaced logarithmically instead of linearly to achieve equal steps in perceived brightness. *Hue* refers to the dominant wavelength in the spectrum and represents the different colors. *Saturation* describes the amount of white light present in the spectrum. If no white light is present, the saturation is 100%. Saturation distinguishes colorful tones from pastel tones at the same hue. In the color cone of Figure 1.2, equal distances between colors by no means correspond to equal perceptual differences. The Commission Internationale de l’Eclairage (CIE) has defined perceptually more uniform color spaces like $L^*u^*v^*$ and $L^*a^*b^*$. A discussion of pros and cons of different color spaces is beyond the scope of this textbook.

While chromatic light needs three descriptors or numbers to characterize its color, achromatic light, as produced by a black-and-white monitor, has only one descriptor, its brightness or *gray value*. Achromatic light is light with a saturation of 0%. It contains only white light.

Chapter 1: Introduction to Digital Image Processing

Given a set of possible gray levels or colors and a (rectangular) grid, a *digital image* attributes a gray value (i.e., brightness) or a color (i.e., hue, saturation, and brightness) to each of the grid points or *pixels*. In a digital image, the gray levels are integers. Although brightness values are continuous in real life, in a digital image we have only a limited number of gray levels at our disposal. The conversion from analog samples to discrete-valued samples is called *quantization*. Figure 1.3 shows the same image using two different quantizations. When too few gray values are used, *contouring* appears. The image is reduced to an artificial looking height map. In order not to lose information by quantizing the detected signal, many more gray values are needed than what is strictly necessary to avoid contouring artifacts. Most digital medical images today use 4096 gray values (12 bpp). The problem with so many gray values is that

subsequent values are not distinguishable anymore on the display. One way to overcome this problem is by expanding a small gray value interval into a larger one with a suitable gray value transformation, as discussed in Section 1.3.1 below.

In the process of digital imaging, the continuous looking world has to be captured onto the finite number of pixels of the image grid. The conversion from a continuous function to a discrete function, retaining only the values at the grid points, is called *sampling* and is discussed in detail in Appendix A.

Much information about an image is contained in its *histogram*. The histogram h of an image is a probability distribution on the set of possible gray levels. The probability of a gray value v is given by its relative frequency in the image, that is,

$$h(v) = \frac{\text{Number of pixels having gray value } v}{\text{Total number of pixels}}. \quad (1.1)$$

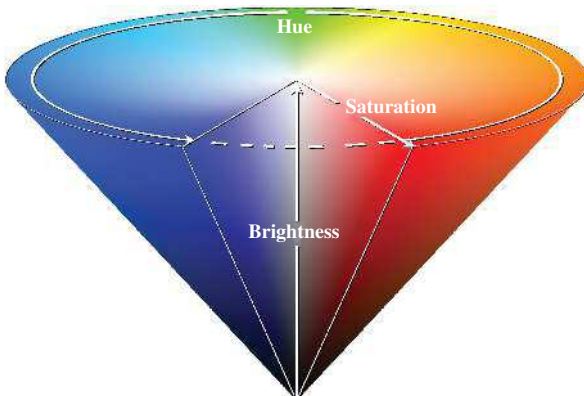


Figure 1.2 Hue, brightness, and saturation. (Reprinted with permission of The MathWorks Inc.)

1.2 Image Quality

The *resolution* of a digital image is sometimes wrongly defined as the linear pixel density (expressed in dots per inch). This is, however, only an upper bound for the resolution. Resolution is also determined by the imaging process. The more blurring, the lower is the resolution. Factors that contribute to the unsharpness of a perceived image are the characteristics of the imaging system, such as the focal spot, the amount of detector blur, and the viewing conditions.

Resolution can be defined as follows. When imaging a very small, bright point on a dark background, this dot will normally not appear as sharp in the image as it actually is. It will be smoothed, and the obtained

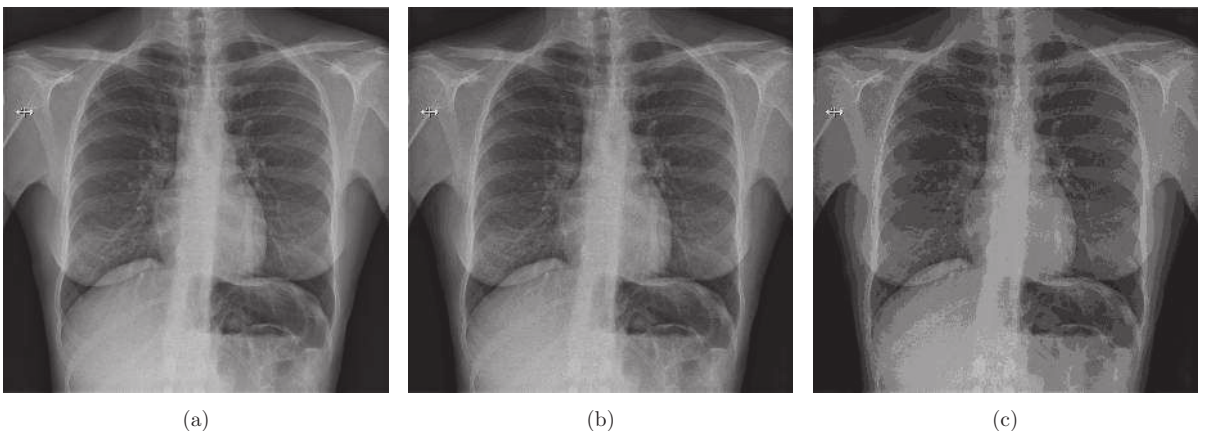


Figure 1.3 Image quantized with (a) 8 bpp, (b) 4 bpp, and (c) 3 bpp.

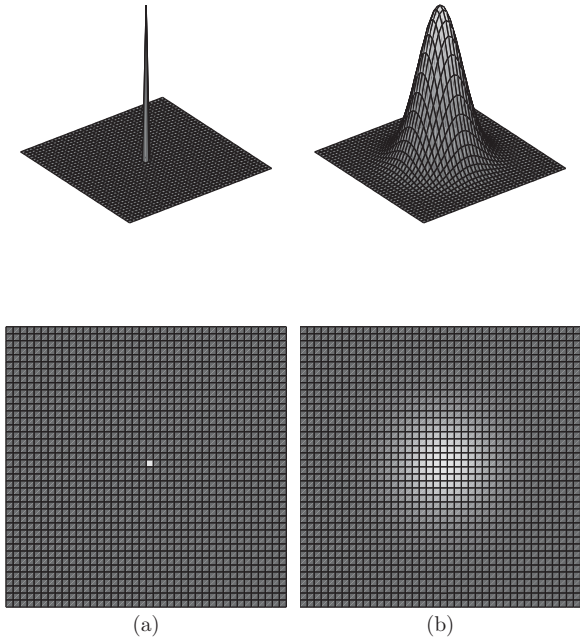


Figure 1.4 (a) Sharp bright spot on a dark background. (b) Typical image of (a). The smoothed blob is called the point spread function (PSF) of the imaging system.

blob is called the *point spread function* (PSF) (see Figure 1.4). An indicative measure of the resolution is the *full width at half maximum* (FWHM) of the point spread function. When two such smoothed blobs are placed at this or a shorter distance from each other, they will no longer be distinguishable as two separate objects. If the resolution is the same in all directions, the *line spread function* (LSF), i.e., the actual image of a thin line, may be more practical than the PSF.

Instead of using the PSF or LSF it is also possible to use the *optical transfer function* (OTF) (see Figure 1.5). The OTF expresses the relative amplitude and phase shift of a sinusoidal target as a function of frequency. The modulation transfer function (MTF) is the amplitude (i.e., $MTF = |OTF|$) and the phase transfer function (PTF) is the phase component of the OTF. For small amplitudes, the lines may no longer be distinguishable. An indication of the resolution is the number of line pairs per millimeter (lp/mm) at a specified small amplitude (e.g., 10%).

As explained in Appendix A, the OTF is the Fourier transform (FT) of the PSF or LSF.

Contrast is the difference in signal intensity of adjacent regions of the image. The contrast is defined by (1) the imaging process, such as the source intensity and the absorption efficiency or sensitivity of the

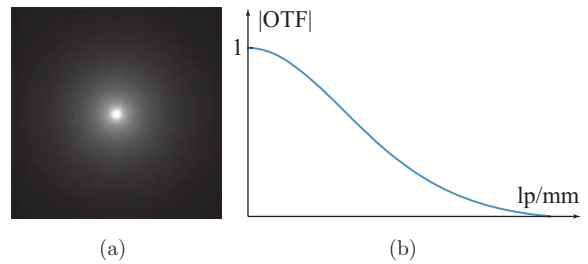


Figure 1.5 (a) Point spread function (PSF). (b) Corresponding modulation transfer function (MTF). The MTF is the amplitude of the optical transfer function (OTF), which is the Fourier transform (FT) of the PSF.

capturing device, (2) the scene characteristics, such as the physical properties, size of the object, and the use of contrast agents, and (3) the viewing conditions, such as the room illumination and display equipment. Remember that the OTF expresses the loss of contrast for increasing frequencies. Because the OTF drops off for larger frequencies, the contrast of very small objects will be influenced by the resolution as well.

A third quality factor is image *noise*. The emission and detection of light and all other types of electromagnetic waves are stochastic processes. Because of the statistical nature of imaging, noise is always present. It is the random component in the image. If the noise level is high compared with the image intensity of an object, the meaningful information is lost in the noise. An important measure, obtained from signal theory, is therefore the *signal-to-noise ratio* (SNR or S/N). For images the *contrast-to-noise ratio* (CNR) is often used, where contrast corresponds to signal difference between adjacent regions. Both contrast and noise are frequency dependent. An estimate of the noise can be obtained by making a *flat-field* image, i.e., an image without an object between the source and the detector.*

Artifacts are artificial image features such as dust or scratches in photographs. Examples in medical images are metal streak artifacts in computed tomography (CT) images and geometric distortions in magnetic resonance (MR) images. Artifacts may also be introduced by digital image processing, such as edge enhancement. Because artifacts may hamper the diagnosis or yield incorrect measurements, it is

* The noise amplitude as a function of spatial frequency can be calculated from the square root of the noise power spectrum (NPS), also called the Wiener spectrum, which is the Fourier transform of the autocorrelation of a flat-field image.

Chapter 1: Introduction to Digital Image Processing

important to avoid them or at least understand their origin.

In the following chapters, image resolution, contrast, noise, and artifacts are discussed for each of the imaging modalities.

1.3 Basic Image Operations

In this section, a number of basic mathematical operations on images are described. They can be employed for image enhancement, analysis, and visualization.

The aim of medical image enhancement is to allow the clinician to better perceive all the relevant diagnostic information present in the image. In digital radiography, for example, 12-bit images with 4096 possible gray levels are available. As discussed above, it is physically impossible for the human eye to distinguish all these gray values at once in a single image. Consequently, not all the diagnostic information encoded in the image may be perceived. Meaningful details must have a sufficiently high contrast to allow the clinician to detect them easily.

The larger the number of gray values in the image, the more important this issue becomes, as lower contrast features may become available in the image data. Therefore, image enhancement will not become less important as the quality of digital image capturing systems improves. On the contrary, it will gain importance.

1.3.1 Gray Level Transformations

Given a digital image I that attributes a gray value (i.e., brightness) to each of the pixels (i, j) , a *gray level*

transformation is a function g that transforms each gray level $I(i, j)$ to another value $I'(i, j)$ independent of the position (i, j) . Hence, for all pixels (i, j)

$$I'(i, j) = g(I(i, j)). \quad (1.2)$$

In practice, g is an increasing function. Instead of transforming gray values, it is also possible to operate on color (i.e., hue, saturation, and brightness). In that case, three of these transformations are needed to transform colors to colors.

Note that, in this textbook, the notation I is used not only for the physical intensity but also for the gray value (or color), which are usually not identical. The gray value can represent brightness (logarithm of the intensity, see Section 1.1), relative signal intensity, or any other derived quantity. Nevertheless, the terms *intensity* and *intensity image* are loosely used as synonyms for *gray value* and *gray value image*.

If pixel (i_1, j_1) appears brighter than pixel (i_2, j_2) in the original image, this relation holds after the gray level transformation. The main use of such a gray level transformation is to increase the contrast in some regions of the image. The price to be paid is a decreased contrast in other parts of the image. Indeed, in a region containing pixels with gray values in the range where the slope of g is larger than 1, the difference between these gray values increases. In regions with gray values in the range with slope smaller than 1, gray values come closer together and different values may even become identical after the transformation. Figure 1.6 shows an example of such a transformation.

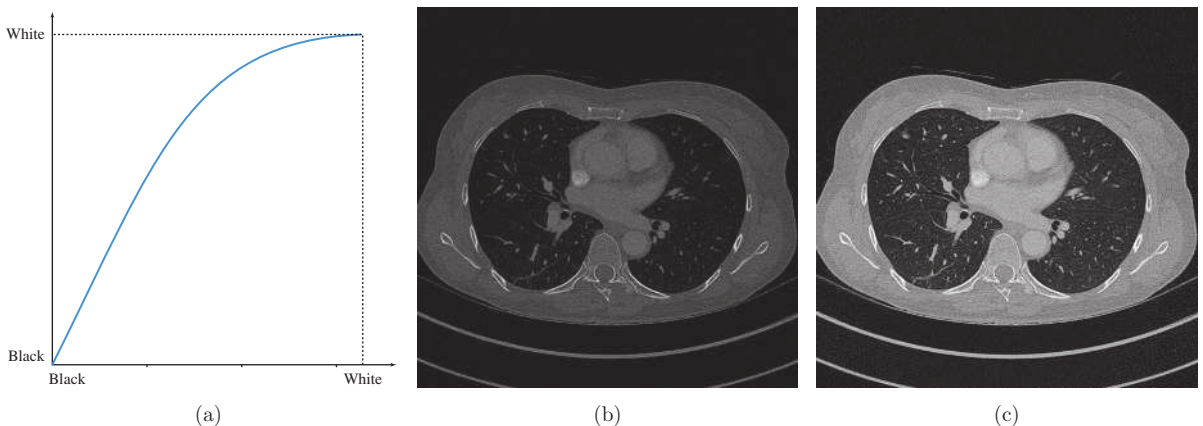


Figure 1.6 (a) A gray level transformation that increases the contrast in dark areas and decreases it in bright regions. It can be used when the clinically relevant information is situated in the dark areas, such as the lungs in this example: (b) the original image, (c) the transformed image.

A particular and popular transformation is the *window/level operation* (see Figure 1.7(a)). In this operation, an interval or window is selected, determined by the window center or level l and the window width w . Explicitly

$$\begin{aligned}
 g_{l,w}(t) &= 0 && \text{for } t < l - \frac{w}{2} \\
 &= \frac{M}{w} \left(t - l + \frac{w}{2} \right) && \text{for } l - \frac{w}{2} \leq t \leq l + \frac{w}{2} \\
 &= M && \text{for } t > l + \frac{w}{2},
 \end{aligned} \tag{1.3}$$

where M is the maximal available gray value. Contrast outside the window is lost completely, whereas the portion of the range lying inside the window is stretched to the complete gray value range. This operation is very useful for images with a bimodal histogram (see Figure 1.8).

An even simpler operation is *thresholding* (Figure 1.7(b)). Here all gray levels up to a certain threshold tr are set to zero, and all gray levels above the threshold equal the maximal gray value

$$\begin{aligned}
 g_{tr}(t) &= 0 && \text{for } t \leq tr \\
 &= M && \text{for } t > tr.
 \end{aligned} \tag{1.4}$$

This operation can be applied to CT images for the segmentation of bony structures, which can subsequently be visualized in three dimensions or printed with a 3D printer.

1.3.2 Multi-image Operations

A simple operation is adding or subtracting images in a pixelwise way. For two images I_1 and I_2 , the sum I_+ and the difference I_- are defined as

$$I_+(i, j) = I_1(i, j) + I_2(i, j) \tag{1.5}$$

$$I_-(i, j) = I_1(i, j) - I_2(i, j). \tag{1.6}$$

If these operations yield values outside the available gray value range, the resulting image can be brought back into that range by a linear transformation. The average of n images is defined as

$$I_{av}(i, j) = \frac{1}{n}(I_1(i, j) + \dots + I_n(i, j)). \tag{1.7}$$

Averaging can be useful to decrease the noise in a sequence of images of a motionless object (Figure 1.9). The random noise averages out, whereas the object remains unchanged (if the images match perfectly). This method can also be used for color

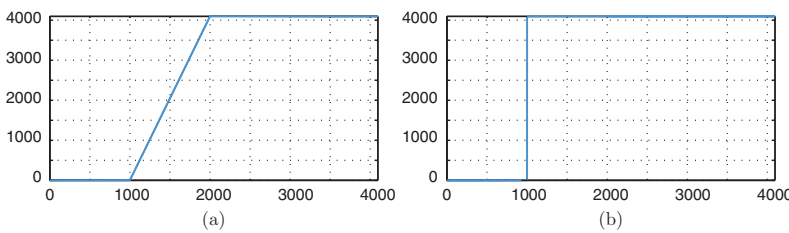


Figure 1.7 (a) Window/leveling with $l = 1500, w = 1000$. (b) Thresholding with $tr = 1000$.

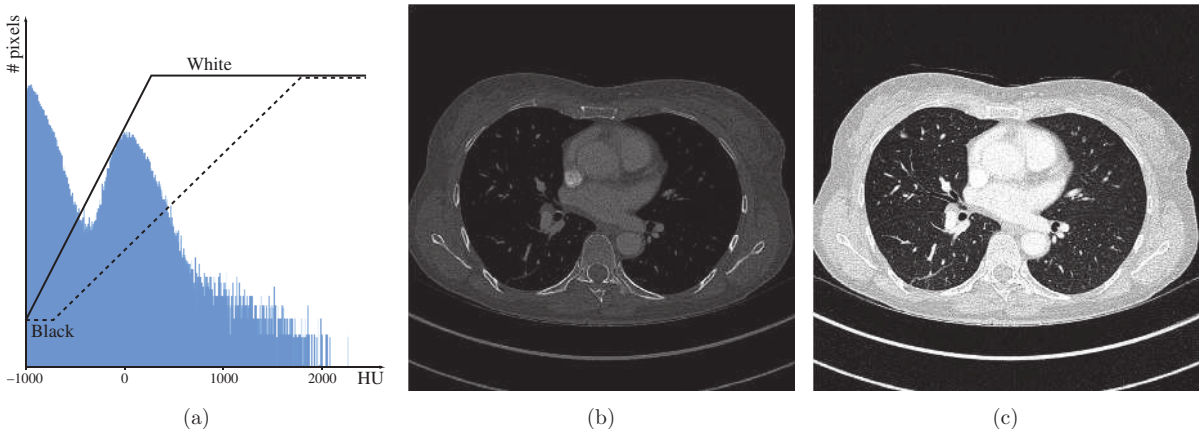


Figure 1.8 (a) Bimodal histogram of the CT image shown in Figure 1.6(b). (b, c) Results of window/leveling with a bone window (dashed line in (a)) and a lung window (solid line in (a)), respectively.

Chapter 1: Introduction to Digital Image Processing

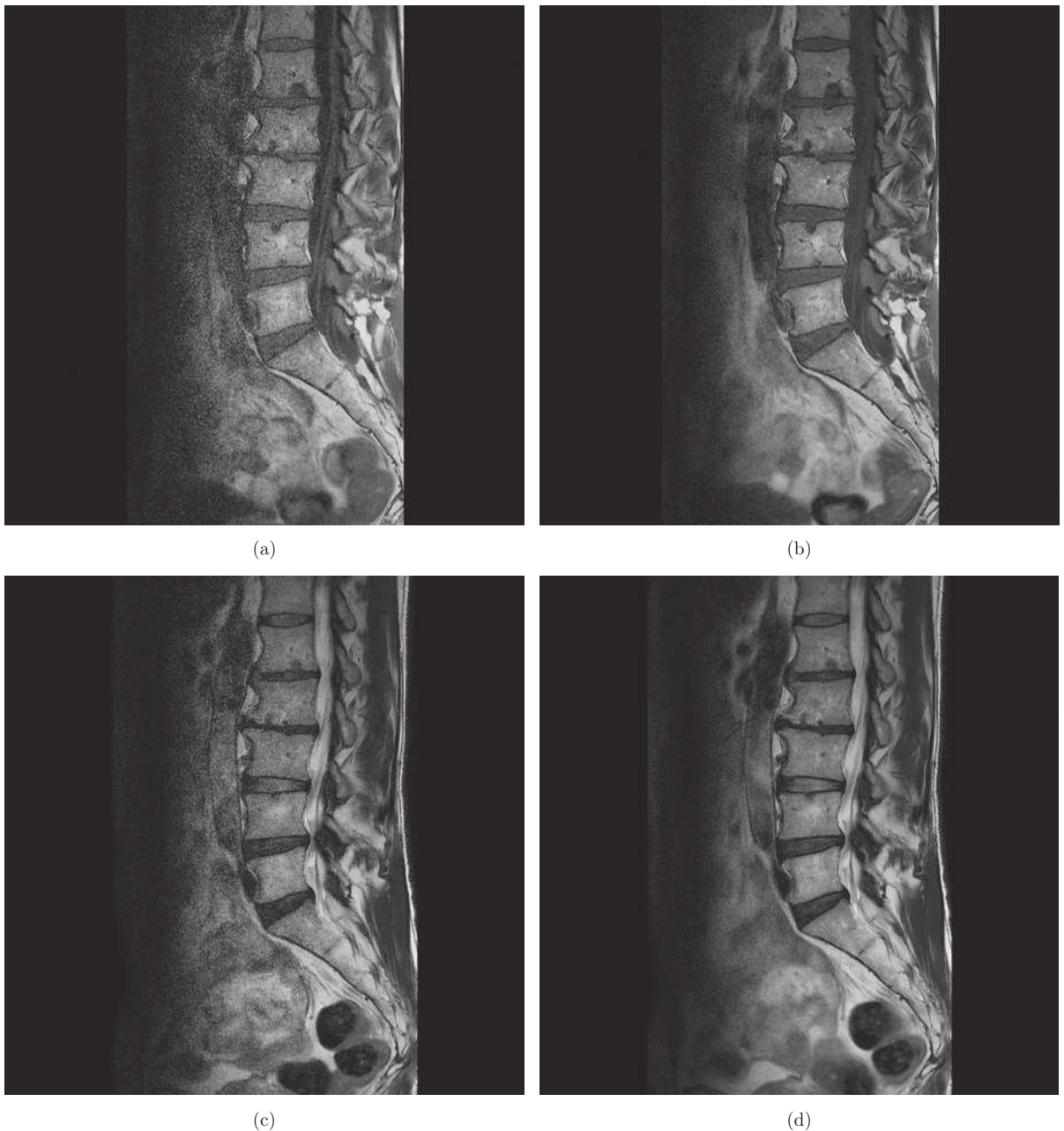


Figure 1.9 Magnetic resonance image of a slice through the spine. Example of averaging to increase the CNR. **(a)** This image was obtained with a T_1 -weighted turboSE sequence (see Section 4.5.8.2). **(b)** Same slice as in **(a)**, obtained by repeating the sequence five times followed by averaging. **(c)** Image obtained with a T_2 -weighted turboSE sequence. **(d)** Averaged image of five subsequent T_2 -weighted sequences. (Courtesy of the Department of Radiology.)

images by averaging the different channels independently like gray level images. Subtraction can be used to get rid of the background in two similar images. For example, in subtraction angiography, two images are made, one with a contrast agent injected

in the blood vessels and one without a contrast agent. Subtraction of these two images yields a pure image of the blood vessels because the subtraction deletes the other anatomical features. Figure 1.10 shows an example.

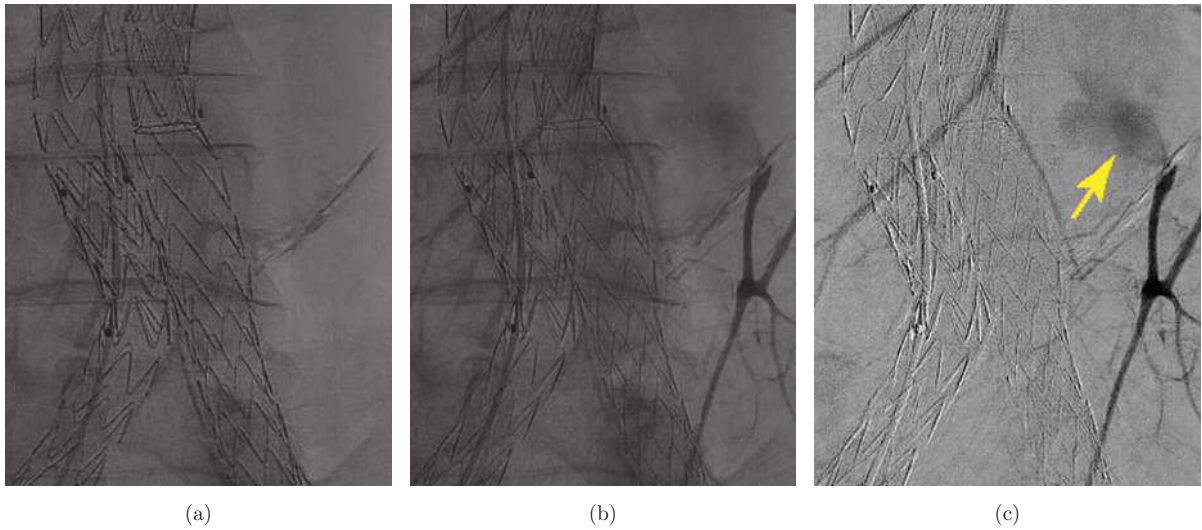


Figure 1.10 (a) Radiographic image showing a stent graft for endovascular aneurysm repair. (b) The same exposure after contrast injection. (c) Subtraction of (b) and (a), followed by contrast enhancement. The subtraction image clearly shows an endoleak, that is, blood flowing into the aneurysm sac via a branch vessel, a common complication after an endovascular aneurysm repair. (Courtesy of Professor G. Maleux, Department of Radiology)

1.3.3 Geometric Operations

It is often necessary to perform elementary geometric operations on an image, such as scaling (zooming), translation, rotation, and shear. Examples are the registration of images (see Section 7.6.1.1) and image-to-patient registration for image-guided surgery (see Section 8.6.2). A spatial or geometric transformation assigns each point (x, y) to a new location $(x', y') = S(x, y)$. The most common two-dimensional (2D) transformations can be written using homogeneous coordinates:

$$\begin{aligned}
 \text{scaling} \quad \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} &= \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \\
 \text{translation} \quad \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} &= \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \\
 \text{shear} \quad \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} &= \begin{pmatrix} 1 & u_x & 0 \\ u_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \\
 \text{rotation} \quad \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} &= \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \\
 \text{general affine} \quad \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} &= \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.
 \end{aligned} \tag{1.8}$$

Composition of two such transformations amounts to multiplying the corresponding matrices.

A general affine 2D transformation depends on six parameters and includes scaling, translation, shear, and rotation as special cases. Affine transformations preserve parallelism of lines but generally not lengths and angles. Angles and lengths are preserved by orthogonal transformations (e.g., rotations and translations)

$$\text{orthogonal} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \tag{1.9}$$

where the 2×2 matrix $R = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix}$ is subject to the constraint $R^T R = 1$.

A pixel $(x, y) = (i, j)$ of image $I(i, j)$ will be mapped onto (x', y') , and x' and y' are usually no longer integer values. To obtain a new image $I'(i', j')$ on a pixel grid, interpolation is used. For each (i', j') the gray value $I'(i', j')$ can then be calculated by simple (e.g., bilinear) interpolation between the gray values of the pixels of I lying closest to the inverse transformation of (i', j') , i.e., $S^{-1}(i', j')$.

Today the majority of medical images are three dimensional (3D). The above matrices can easily be extended to three dimensions. For example, the general affine 3D transformation can be written as

Chapter 1: Introduction to Digital Image Processing

$$\text{general affine} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (1.10)$$

While most medical images are three dimensional, interventional imaging is often still two dimensional. To map 3D image data onto a 2D plane, a projective transformation is needed. Assuming a pinhole camera, such as an X-ray tube, with focal point $(0, 0, 0)$, any projection line through (x, y, z) intersects the projection plane $z = f$ in the point (x_p, y_p) as follows

$$\begin{aligned} \frac{x}{x_p} &= \frac{z}{f} \\ \frac{y}{y_p} &= \frac{z}{f}. \end{aligned} \quad (1.11)$$

This projection can be written in matrix form as follows:

$$\begin{pmatrix} \lambda \cdot x_p \\ \lambda \cdot y_p \\ \lambda \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (1.12)$$

with λ being an arbitrary value.

Using homogeneous coordinates, the above geometric transformations can all be represented by matrices. In some cases, however, it can be necessary to use more flexible transformations. For example, the superposition of images in dynamic or follow-up studies may be hampered by patient movement and tissue deformation; the spatial alignment of images of different patients in population studies has to cope with intra-patient shape variability; and undesired deviations of the magnetic field in magnetic resonance imaging (see p. 107) can cause complex geometric distortions.

1.3.4 Filters

1.3.4.1 Linear Filters

From linear system theory (see Eq. A.22), we know that an image $I(i, j)$ can be written as follows:

$$I(i, j) = \sum_{k,l} I(k, l)\delta(i - k, j - l). \quad (1.13)$$

For a linear shift-invariant transformation \mathcal{L} (see also Eq. A.31),

$$\begin{aligned} \mathcal{L}(I)(i, j) &= \sum_{k,l} I(k, l)\mathcal{L}(\delta)(i - k, j - l) \\ &= \sum_{k,l} I(k, l)f(i - k, j - l) \\ &= \sum_{k,l} f(k, l)I(i - k, j - l) \\ &= f * I(i, j), \end{aligned} \quad (1.14)$$

where f is called the *kernel* or *filter*, and the linear transformation on the digital image I is the discrete convolution with its kernel $f = \mathcal{L}(\delta)$.

In practice, the flipped kernel h defined as $h(i, j) = f(-i, -j)$ is usually used. Hence, Eq.(1.14) can be rewritten as

$$\begin{aligned} \mathcal{L}(I)(i, j) &= f * I(i, j) \\ &= \sum_{k,l} f(k, l)I(i - k, j - l) \\ &= \sum_{k,l} h(k, l)I(i + k, j + l) \\ &= h \bullet I(i, j), \end{aligned} \quad (1.15)$$

where $h \bullet I$ is the *cross-correlation* of h and I . If the filter is symmetric, which is often the case, cross-correlation and convolution are identical.

A cross-correlation of an image $I(i, j)$ with a kernel h has the following physical meaning. The kernel h is used as an image *template* or *mask* that is shifted across the image. For every image pixel (i, j) , the template pixel $h(0, 0)$, which typically lies in the center of the mask, is superimposed onto this pixel (i, j) , and the values of the template and image that correspond to the same positions are multiplied. Next, all these values are summed. A cross-correlation emphasizes patterns in the image similar to the template.

Often local filters with only a few pixels in diameter are used. A simple example is the 3×3 mask with values $1/9$ at each position (Figure 1.11). This filter performs an averaging on the image, making it smoother and removing some noise. The filter gives the same weight to the center pixel as to its neighbors. A softer way of smoothing the image is to give a high weight to the center pixel and less weight to pixels further away from the central pixel. A suitable filter for this operation is a sampled Gaussian kernel

$$g(\vec{r}) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-r^2}{2\sigma^2}\right) \quad \vec{r} = (i, j). \quad (1.16)$$

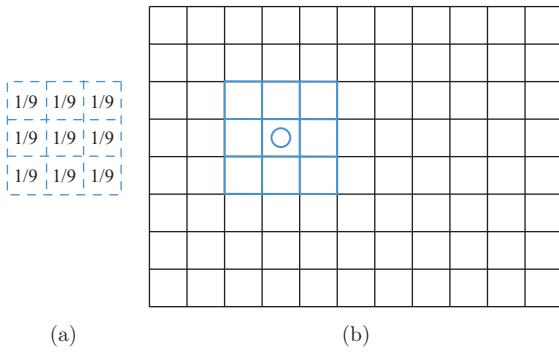


Figure 1.11 (a) 3×3 averaging filter. (b) The filter as floating image template or mask.

Small values are put to zero in order to produce a local filter. The Fourier transform of the Gaussian is again Gaussian. In the Fourier domain, convolution with a filter becomes multiplication (see Appendix A). Taking this into account, it is clear that a Gaussian filter attenuates the high frequencies in the image. These averaging filters are therefore also called *low-pass* filters. In contrast, filters that emphasize high frequencies are called *high-pass* filters. A high-pass filter can be constructed simply from a low-pass one by subtracting the low-pass filter g from the identity filter δ . A high-pass filter enhances small-scale variations in the image. It extracts edges and fine textures. An example of low-pass and high-pass filtering is shown in Figure 1.12.

Another type of linear filters are differential operators such as the gradient and the Laplacian. However, these operations are not defined on discrete images. Because derivatives are defined on differentiable functions, the computation is performed by first fitting a differentiable function through the discrete dataset. This can be obtained by convolving the discrete image with a continuous function f . The derivative of this result is evaluated at the points (i, j) of the original sampling grid. For the 1D partial derivative, this sequence of operations can be written as follows:

$$\begin{aligned} \frac{\partial}{\partial x} I(i, j) &\approx \left[\frac{\partial}{\partial x} \left(\sum_{k,l} I(k, l) f(x - k, y - l) \right) \right]_{x=i, y=j} \\ &= \left[\sum_{k,l} \frac{\partial f}{\partial x} (i - k, j - l) I(k, l) \right]. \end{aligned} \quad (1.17)$$

Hence, the derivative is approximated by a convolution with a filter that is the sampled derivative of some differentiable function $f(\vec{r})$. This procedure can now be used further to approximate the gradient and the Laplacian of a digital image:

$$\begin{aligned} \nabla I &= \nabla f * I \\ \nabla^2 I &= \nabla^2 f * I, \end{aligned} \quad (1.18)$$

where it is understood that we use the discrete convolution. If f is a Gaussian g , the following differential convolution operators are obtained:

$$\begin{aligned} \nabla g(\vec{r}) &= -\frac{1}{\sigma^2} g(\vec{r}) \cdot \vec{r} \\ \nabla^2 g(\vec{r}) &= \frac{1}{\sigma^4} (r^2 - 2\sigma^2) \cdot g(\vec{r}). \end{aligned} \quad (1.19)$$

For $\sigma = 0.5$, this procedure yields approximately the following 3×3 filters (see Figure 1.13):

gaussian	0.01	0.08	0.01
	0.08	0.64	0.08
	0.01	0.08	0.01
$\frac{\partial}{\partial x}$	0.05	0	-0.05
	0.34	0	-0.34
	0.05	0	-0.05
$\frac{\partial}{\partial y}$	0.05	0.34	0.05
	0	0	0
	-0.05	-0.34	-0.05
∇^2	0.3	0.7	0.3
	0.7	-4	0.7
	0.3	0.7	0.3

Note that integration of a Gaussian over the whole spatial domain must be 1, and for the gradient and Laplacian this must be 0. To satisfy this condition, the numbers in the templates above, which are spatially limited, were adapted.

The Laplacian of a Gaussian (LoG) is often approximated as a difference of Gaussians (DoG) with different values of σ . This can be derived from the partial derivative of a Gaussian function with respect to σ , which can be written as

$$\frac{\partial g(\vec{r})}{\partial \sigma} = \sigma \nabla^2 g(\vec{r}). \quad (1.20)$$

Chapter 1: Introduction to Digital Image Processing

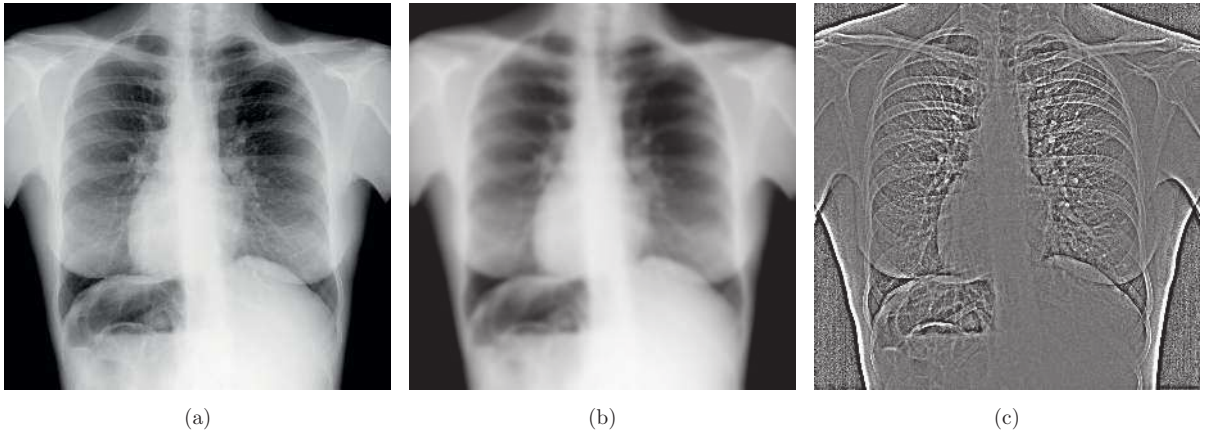


Figure 1.12 (a) Chest radiography. (b) Low-pass filtered image with a Gaussian filter ($\sigma = 2.2$ mm). (c) High-pass filtered image obtained by subtracting (b) from (a).

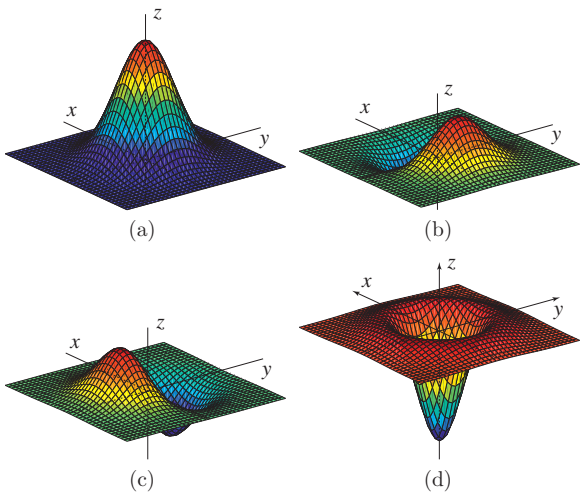


Figure 1.13 (a) A Gaussian function. (b) Derivative of the Gaussian in the x -direction. (c) Derivative of the Gaussian in the y -direction. (d) Laplacian of the Gaussian.

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

average - δ	1	1	1
	1	-8	1
	1	1	1

Note that if we compute the convolution of an image with a filter, it is necessary to extend the image at its boundaries because pixels lying outside the image will be addressed by the convolution algorithm. This is best done in a smooth way, for example, by repeating the boundary pixels. If not, artifacts appear at the boundaries after the convolution.

As an application of linear filtering, let us discuss edge enhancement using *unsharp masking*. Figure 1.14 shows an example. As already mentioned, a low-pass filter g can be used to split an image I into two parts: a smooth part $g * I$ and the remaining high-frequency part $I - g * I$, containing the edges in the image or image details. Hence

$$I = g * I + (I - g * I). \tag{1.23}$$

Note that $I - g * I$ is a crude approximation of the Laplacian of I . Unsharp masking enhances the image details by emphasizing the high-frequency part and assigning it a higher weight. For some $\alpha > 0$, the output image I' is then given by

Approximating this partial derivate as

$$\frac{\partial g(\vec{r})}{\partial \sigma} \approx \frac{g(\vec{r}, k\sigma) - g(\vec{r}, \sigma)}{k\sigma - \sigma}, \tag{1.21}$$

for small-scale factors k , Eq. 1.20 becomes

$$g(\vec{r}, k\sigma) - g(\vec{r}, \sigma) \approx (k - 1)\sigma \nabla^2 g(\vec{r}). \tag{1.22}$$

Popular derivative filters are the Sobel operator for the first derivative, and the *average - δ* for the Laplacian, which use integer filter elements: