3.1

Basic Linear Filtering with Application to Image Enhancement

Alan C. Bovik
The University of Texas at Austin

Scott T. Acton
University of Virginia

1 Introduction

Linear system theory and linear filtering play central roles in digital image and video processing. Many potent techniques for modifying, improving, or representing digital visual data are expressed in terms of linear systems concepts. Linear filters are used for generic tasks such as image/video contrast improvement, denoising, and sharpening, as well as for more object- or feature-specific tasks such as target matching and feature enhancement.

Much of this *Handbook* deals with the application of linear filters to image and video enhancement, restoration, reconstruction, detection, segmentation, compression, and transmission. The goal of this chapter is to introduce some of the basic supporting ideas of linear systems theory as they apply to digital image filtering, and to outline some of the applications. Special emphasis is given to the topic of linear image enhancement.

We will require some basic concepts and definitions in order to proceed. The basic two-dimensional discrete-space signal is the *two-dimensional impulse function,* defined by

$$\delta(m-p,\, n-q) = \begin{cases} 1; & m = p \text{ and } n = q \\ 0; & \text{else} \end{cases} . \qquad (1)$$

Thus, (1) takes unit value at coordinate (p, q) and is everywhere else zero. The function in (1) is often termed the *Kronecker delta function* or the *unit sample sequence* [1]. It plays the same role and has the same significance as the so-called *Dirac delta function* of continuous system theory. Specifically, the response of linear systems to (1) will be used to characterize the general responses of such systems.

Any discrete-space image f may be expressed in terms of the impulse function (1):

$$\begin{aligned} f(m,n) &= \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} f(m-p, n-q)\, \delta(p,q) \\ &= \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} f(p,q)\, \delta(m-p, n-q). \end{aligned} \qquad (2)$$

The expression (2), called the *sifting property,* has two meaningful interpretations here. First, any discrete-space image can be written as a sum of weighted, shifted unit impulses. Each weighted impulse comprises one of the pixels of the image. Second, the sum in (2) is in fact a discrete-space linear convolution. As is apparent, the linear convolution of any image f with the impulse function δ returns the function unchanged.

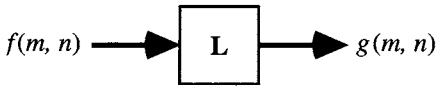


FIGURE 1 Two-dimensional input-output.

The impulse function effectively describes certain systems known as *linear space–invariant (LSI) systems*. We explain these terms next.

A two-dimensional system **L** is a process of image transformation, as shown in Fig. 1:

We can write

$$g(m,n) = \mathbf{L}[f(m,n)]. \tag{3}$$

The system **L** is *linear* if and only if for any $f_1(m,n)$, $f_2(m,n)$ such that

$$g_1(m,n) = \mathbf{L}[f_1(m,n)] \quad \text{and} \quad g_2(m,n) = \mathbf{L}[f_2(m,n)] \tag{4}$$

and any two constants a, b, then

$$a \cdot g_1(m,n) + b \cdot g_2(m,n) = \mathbf{L}[a \cdot f_1(m,n) + b \cdot f_2(m,n)] \tag{5}$$

for every (m, n). This is often called the *superposition property* of linear systems.

The system **L** is shift-invariant if for every $f(m,n)$ such that (3) holds, then also

$$g(m-p,n-q) = \mathbf{L}[f(m-p,n-q)] \tag{6}$$

for any (p, q). Thus, a spatial shift in the input to **L** produces no change in the output, except for an identical shift.

The rest of this chapter will be devoted to studying systems that are linear and shift-invariant (LSI). In this and other chapters, it will be found that LSI systems can be used for many powerful image and video processing tasks. In yet other chapters, nonlinearity and/or space-variance will be shown to afford certain advantages, particularly in surmounting the inherent limitations of LSI systems.

2 Impulse Response, Linear Convolution, and Frequency Response

The *unit impulse response* of a two-dimensional input–output system **L** is:

$$\mathbf{L}[\delta(m-p,n-q)] = h(m,n;p,q). \tag{7}$$

This is the response of system **L**, at spatial position (m, n), to an impulse located at spatial position (p, q). Generally, the impulse response is a function of these four spatial variables. However, if the system **L** is space-invariant, then if

$$\mathbf{L}[\delta(m,n)] = h(m,n) \tag{8}$$

is the response to an impulse applied at the spatial origin, then also

$$\mathbf{L}[\delta(m-p,n-q)] = h(m-p,n-q), \tag{9}$$

which means that the response to an impulse applied at any spatial position can be found from the impulse response (8).

As already mentioned, the discrete-space impulse response $h(m, n)$ completely characterizes the input–output response of LSI input–output systems. This means that if the impulse response is known, an expression can be found for the response to any input. The form of the expression is two-dimensional discrete-space linear convolution.

Consider the generic system **L** shown in Fig. 1, with input $f(m, n)$ and output $g(m, n)$. Assume that the response is due to the input f only (the system would be at rest without the input). Then from (2):

$$g(m,n) = \mathbf{L}[f(m,n)] = \mathbf{L}\left[\sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} f(p,q)\delta(m-p,n-q)\right] \tag{10}$$

If the system is known to be linear, then

$$g(m,n) = \sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} f(p,q)\,\mathbf{L}[\delta(m-p,n-q)] \tag{11}$$

$$= \sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} f(p,q)\,h(m,n;p,q), \tag{12}$$

which is all that generally can be said without further knowledge of the system and the input. If it is known that the system is space-invariant (hence LSI), then (12) becomes

$$g(m,n) = \sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} f(p,q)\,h(m-p,n-q) \tag{13}$$

$$= f(m,n)^{*}h(m,n), \tag{14}$$

which is the two-dimensional discrete space linear convolution of input f with impulse response h.

The linear convolution expresses the output of a wide variety of electrical and mechanical systems. In continuous

systems, the convolution is expressed as an integral. For example, with lumped electrical circuits, the convolution integral is computed in terms of the passive circuits elements (resistors, inductors, capacitors). In optical systems, the integral uses the point-spread functions of the optics. The operations occur effectively instantaneously, with the computational speed limited only by the speed of the electrons or photons through the system elements.

However, in discrete signal and image processing systems, the discrete convolutions are calculated sums of products. This convolution can be directly evaluated at each coordinate (m, n) by a digital processor, or, as discussed in Chapter 2.3, it can be computed using the discrete Fourier transform (DFT) using a fast Fourier transform (FFT) algorithm. Of course, if the exact linear convolution is desired, this means that the involved functions must be appropriately *zero-padded* prior to using the DFT, as discussed in Chapter 2.3. The DFT/FFT approach is usually, but not always faster. If an image is being convolved with a very small spatial filter, then direct computation of (14) can be faster.

Suppose that the input to a discrete LSI system with impulse response $h(m, n)$ is a complex exponential function:

$$\begin{aligned} f(m,n) &= e^{2\pi j(Um+Vn)} \\ &= \cos[2\pi(Um+Vn)] + j\sin[2\pi(Um+Vn)]. \end{aligned} \tag{15}$$

Then the system response is the linear convolution:

$$g(m,n) = \sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} h(p,q)f(m-p,n-q)$$

$$= \sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} h(p,q)e^{2\pi j[U(m-p)+V(n-q)]} \tag{16}$$

$$= e^{2\pi j(Um+Vn)}\sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} h(p,q)e^{-2\pi j(Up+Vq)}, \tag{17}$$

which is exactly the input $f(m, n) = e^{2\pi j(Um+Vn)}$ multiplied by a function of (U, V) only:

$$\begin{aligned} H(U,V) &= \sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} h(p,q)e^{-2\pi j(Up+Vq)} \\ &= |H(U,V)|\cdot e^{j\angle H(U,V)}. \end{aligned} \tag{18}$$

The function $H(U, V)$, which is immediately identified as the discrete-space Fourier transform (or DSFT, discussed extensively in Chapter 2.3) of the system impulse response, is called the *frequency response* of the system.

From (17) it may be seen that the response to any complex exponential sinusoid function, with frequencies (U, V), is the same sinusoid, but with its amplitude scaled by the system *magnitude response* $|H(U, V)|$ evaluated at (U, V) and with a shift equal to the system *phase response* $\angle H(U, V)$ at (U, V). The complex sinusoids are the unique functions that have this invariance property in LSI systems.

As mentioned, the impulse response $h(m, n)$ of an LSI system is sufficient to express the response of the system to any input.[1] The frequency response $H(U, V)$ is uniquely obtainable from the impulse response (and vice versa), and so contains sufficient information to compute the response to any input that has a DSFT. In fact, the output can be expressed in terms of the frequency response via $G(U, V) = F(U, V)H(U, V)$ and via the DFT/FFT with appropriate zero-padding. In fact, throughout this chapter and elsewhere, it may be assumed that whenever a DFT is being used to compute linear convolution, the appropriate zero-padding has been applied to avoid the wraparound effect of the cyclic convolution.

Usually, linear image processing filters are characterized in terms their frequency responses, specifically by their spectrum-shaping properties. Coarse common descriptions that apply to many two-dimensional image processing include low pass, band pass, or high pass. In such cases, the frequency response is primarily a function of radial frequency and may even be circularly symmetric, namely, a function of $U^2 + V^2$ only. In other cases, the filter may be strongly directional or oriented, with response strongly depending on the frequency angle of the input. Of course, the terms *low pass, band pass, high pass,* and *oriented* are only rough qualitative descriptions of a system frequency response. Each broad class of filters has some generalized applications. For example, low-pass filters strongly attenuate all but the "lower" radial image frequencies (as determined by some bandwidth or cutoff frequency), and so are primarily smoothing filters. They are commonly used to reduce high-frequency noise, eliminate all but coarse image features, or reduce the bandwidth of an image before transmission through a low-bandwidth communication channel or before subsampling the image (see Chapter 7.1).

A (radial frequency) band-pass filter attenuates all but an intermediate range of "middle" radial frequencies. This is commonly used for the enhancement of certain image features, such as edges (sudden transitions in intensity) or the ridges in a fingerprint. A high-pass filter attenuates all but the "higher" radial frequencies, or commonly, significantly amplifies high frequencies without attenuating lower frequencies. This approach is often used for correcting images that are blurred (see Chapter 3.5).

Oriented filters, which attenuate frequencies falling outside of a narrow range of orientations or amplify a narrow range of angular frequencies, tend to be more specialized. For example,

[1] Strictly speaking, for any bounded input, and provided that the system is stable. In practical image processing systems, the inputs are invariably bounded. Also, almost all image processing filters do not involve feedback, and hence are naturally stable.

it may be desirable to enhance vertical image features as a prelude to detecting vertical structures, such as buildings.

Of course, filters may be a combination of types, such as band pass and oriented. In fact, such filters are the most common types of basis functions used in the powerful wavelet image decompositions (Chapters 4.2) that have recently found so many applications in image analysis (Chapter 4.4), human visual modeling (Chapter 4.1), and image and video compression (Chapters 5.4 and 6.2).

In the remainder of this chapter, we introduce the simple but important application of linear filtering for *linear image enhancement,* which specifically means attempting to smooth image noise while not disturbing the original image structure.[2]

3 Linear Image Enhancement

The term *enhancement* implies a process whereby the visual quality of the image is improved. However, the term *image enhancement* has come to specifically mean a process of smoothing irregularities or noise that has somehow corrupted the image, while modifying the original image information as little as possible. The noise is usually modeled as an additive noise or as a multiplicative noise. We will consider additive noise now. As noted in Chapter 4.5, multiplicative noise, which is the other common type, can be converted into additive noise in a homomorphic filtering approach.

Before considering methods for image enhancement, we will make a simple model for additive noise. Chapter 4.5 of this *Handbook* greatly elaborates image noise models, which prove particularly useful for studying image enhancement filters that are nonlinear.

We will make the practical assumption that an observed noisy image is of finite extent $M \times N$: $f = [f(m, n);\ 0 \le m \le M-1,\ 0 \le n \le N-1]$. We model f as a sum of an original image $\mathbf{o}$ and a *noise image* $\mathbf{q}$:

$$f = \mathbf{o} + \mathbf{q} \tag{19}$$

where $\mathbf{n} = (m, n)$. The additive noise image $\mathbf{q}$ models an undesirable, unpredictable corruption of $\mathbf{o}$. The process $\mathbf{q}$ is called a *two-dimensional random process* or a *random field.* Random additive noise can occur as thermal circuit noise, communication channel noise, sensor noise, and so on. Quite commonly, the noise is present in the image signal before it is sampled, so the noise is also sampled coincident with the image.

[2]The term *image enhancement* has been widely used in the past to describe any operation that improves image quality by some criteria. However, in recent years the meaning of the term has evolved to denote image-preserving noise smoothing. This primarily serves to distinguish it from similar-sounding terms, such as *image restoration* and *image reconstruction,* which also have taken specific meanings.

In (19), both the original image and noise image are unknown. The goal of enhancement is to recover an image $\mathbf{g}$ that resembles $\mathbf{o}$ as closely as possible by reducing $\mathbf{q}$. If there is an adequate model for the noise, then the problem of finding $\mathbf{g}$ can be posed as an *image estimation* problem, where $\mathbf{g}$ is found as the solution to a statistical optimization problem. Basic methods for image estimation are also discussed in Chapter 4.5, and in some of the following chapters on image enhancement using nonlinear filters.

With the tools of Fourier analysis and linear convolution in hand, we will now outline the basic approach of image enhancement by linear filtering. More often than not, the detailed statistics of the noise process $\mathbf{q}$ are unknown. In such cases, a simple linear filter approach can yield acceptable results, if the noise satisfies certain simple assumptions.

We will assume a *zero-mean additive white noise* model. The zero-mean model is used in Chapter 2.1, in the context of frame averaging. The process $\mathbf{q}$ is *zero-mean* if the average or *sample mean* of R arbitrary noise samples

$$\left(\frac{1}{R}\right)\sum_{r=1}^{R} q(m_r, n_r) \to 0 \tag{20}$$

as R grows large (provided that the noise process is mean-ergodic, which means that the sample mean approaches the statistical mean for large samples).

The term *white noise* is an idealized model for noise that has, on the average, a broad spectrum. It is a simplified model for *wideband noise.* More precisely, if $Q(U, V)$ is the DSFT of the noise process $\mathbf{q}$, then Q is also a random process. It is called the *energy spectrum* of the random process $\mathbf{q}$. If the noise process is white, then the average squared magnitude of $Q(U, V)$ takes constant over all frequencies in the range $[-\gamma\pi, \gamma\pi]$. In the ensemble sense, this means that the sample average of the magnitude spectra of R noise images generated from the same source becomes constant for large R:

$$\left(\frac{1}{R}\right)\sum_{r=1}^{R} |Q_r(U, V)| \to \eta \tag{21}$$

for all (U, V) as R grows large. The square $\gamma\eta^2$ of the constant level is called the *noise power.* Because $\mathbf{q}$ has finite extent $M \times N$, it has a DFT $\tilde{\mathbf{Q}} = [\tilde{Q}(u, v):\ 0 \le u \le M-1,\ 0 \le v \le N-1]$. On average, the magnitude of the noise DFT $\tilde{\mathbf{Q}}$ will also be flat. Of course, it is highly unlikely that a given noise DSFT or DFT will actually have a flat magnitude spectrum. However, it is an effective simplified model for unknown, unpredictable broadband noise.

Images are also generally thought of as relatively broadband signals. Significant visual information may reside at mid-to-high spatial frequencies, because visually significant image details such as edges, lines, and textures typically contain

higher frequencies. However, the magnitude spectrum of the image at higher image frequencies is usually low; most of the image power resides in the low frequencies contributed by the dominant luminance effects. Nevertheless, the higher image frequencies are visually significant.

The basic approach to linear image enhancement is low-pass filtering. There are different types of low-pass filters that can be used; several will be studied in the following sections. For a given filter type, different degrees of smoothing can be obtained by adjusting the filter bandwidth. A narrower bandwidth low-pass filter will reject more of the high-frequency content of a white or broadband noise, but it may also degrade the image content by attenuating important high-frequency image details. This tradeoff is difficult to balance.

Next, we describe and compare several smoothing low-pass filters that are commonly used for linear image enhancement.

Moving-Average Filter

The moving-average filter can be described in several equivalent ways. First, using the notion of *windowing* introduced in Chapter 2.2, the moving average can be defined as an algebraic operation performed on local image neighborhoods according to a geometric rule defined by the window. Given an image f to be filtered and a window $\mathbf{B}$ that collects gray-level pixels according to a geometric rule (defined by the window shape), then the moving-average–filtered image $\mathbf{g}$ is given by

$$\mathbf{g}(\mathbf{n}) = \mathrm{AVE}[\mathbf{B}f(\mathbf{n})] \tag{22}$$

where the operation AVE computes the sample average of its. Thus, the local average is computed over each local neighborhood of the image, producing a powerful smoothing effect. The windows are usually selected to be symmetric, as with those used for binary morphologic image filtering (Chapter 2.2).

Since the average is a linear operation, it is also true that

$$\mathbf{g}(\mathbf{n}) = \mathrm{AVE}[\mathbf{B}o(\mathbf{n})] + \mathrm{AVE}[\mathbf{B}q(\mathbf{n})]. \tag{23}$$

Because the noise process $\mathbf{q}$ is assumed to be zero-mean in the sense of (20), then the last term in (23) will tend to zero as the filter window is increased. Thus, the moving-average filter has the desirable effect of reducing zero-mean image noise towards zero. However, the filter also effects the original image information. It is desirable that $\mathrm{AVE}[\mathbf{B}o(\mathbf{n})] \approx o(\mathbf{n})$ at each $\mathbf{n}$, but this will not be the case everywhere in the image if the filter window is too large. The moving-average filter, which is low-pass, will blur the image, especially as the window span is increased. Balancing this tradeoff is often a difficult task.

The moving-average filter operation (22) is actually a linear convolution. In fact, the impulse response of the filter is defined as having value $1/R$ over the span covered by the window when centered at the spatial origin (0, 0), and zero elsewhere, where R is the number of elements in the window.

For example, if the window is SQUARE $[(2P+1)^2]$, which is the most common configuration (it is defined in Chapter 2.2), then the average filter impulse response is given by

$$h(m,n) = \begin{cases} 1/(2P+1)^2 & ; -P \le m,n \le P \\ 0 & ; \text{else} \end{cases}. \tag{24}$$

The frequency response of the moving-average filter (24) is:

$$H(U,V) = \frac{\sin[(2P+1)\pi U]}{(2P+1)\sin(\pi U)} \cdot \frac{\sin[(2P+1)\pi V]}{(2P+1)\sin(\pi V)}. \tag{25}$$

The *half-peak bandwidth* is often used for image processing filters. The half-peak (or 3 dB) cutoff frequencies occur on the locus of points (U, V) where $|H(U, V)|$ falls to ½. For the filter (25), this locus intersects the U-axis and V-axis at the cutoffs $U_{\text{half-peak}}$, $V_{\text{half-peak}} \approx 0.6/(2P+1)$ cycles/pixel.

As depicted in Fig. 2, the magnitude response $|H(U, V)|$ of the filter (25) exhibits considerable *sidelobes*. In fact, the number of sidelobes in the range $[0, \gamma\pi]$ is P. As P is increased, the filter bandwidth naturally decreases (more high-frequency attenuation or smoothing), but the overall sidelobe energy does not. The sidelobes are in fact a significant drawback, because there is considerable *noise leakage* at high noise frequencies. These residual noise frequencies remain to degrade the image. Nevertheless, the moving-average filter has been commonly used because of its general effectiveness in the sense of (21) and because of its simplicity (ease of programming).

The moving-average filter can be implemented either as a direct two-dimensional convolution in the space domain, or using DFTs to compute the linear convolution (see Chapter 2.3).

Because application of the moving-average filter balances a tradeoff between noise smoothing and image smoothing, the filter span is usually taken to be an intermediate value. For images of the most common sizes (e.g., 256×256 or 512×512), typical (SQUARE) average filter sizes range from 3×3 to 15×15. The upper end provides significant (and probably excessive) smoothing, because 225 image samples are being averaged to produce each new image value. Of course, if an image suffers from severe noise, then a larger window might be used. A large window might also be acceptable if it is known that the original image is very smooth everywhere.

Figure 3 depicts the application of the moving-average filter to an image that has had zero-mean white Gaussian noise added to it. In the current context, the distribution (Gaussian) of the noise is not relevant, although the meaning can be found in Chapter 4.5. The original image is included for comparison. The image was filtered with SQUARE-shaped moving-average filters of window sizes 5×5 and 9×9,

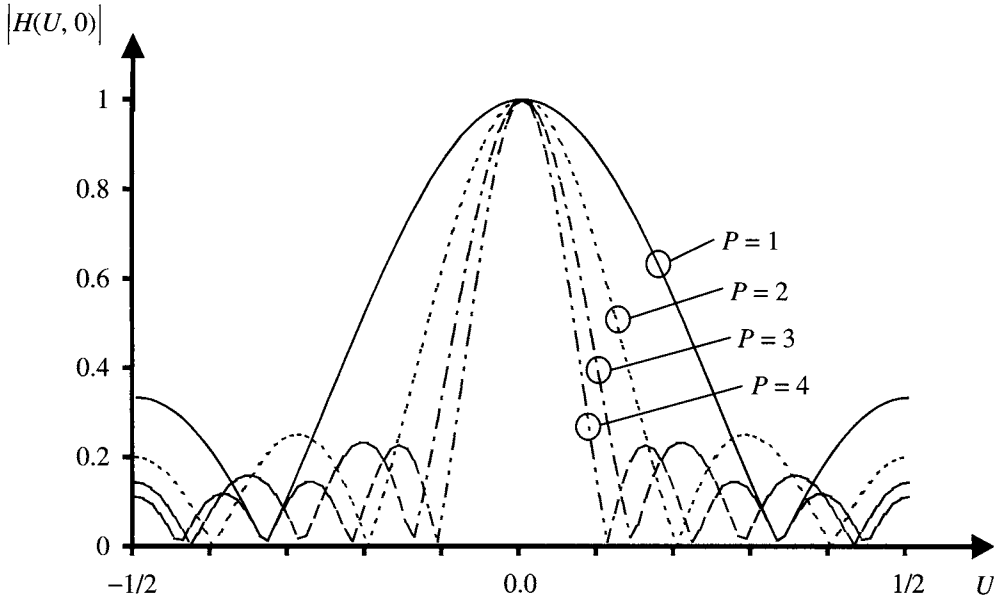


FIGURE 2 Plots of $|H(U,V)|$ given in (25) along $V = 0$, for $P = 1, 2, 3, 4$. As the filter span is increased, the bandwidth decreases. The number of sidelobes in the range $[0, \pi]$ is P.

producing images with significantly different appearances from each other as well as the noisy image. With the 5×5 filter, the noise is inadequately smoothed, yet the image has been blurred noticeably. The result of the 9×9 moving-average filter is much smoother, although the noise influence is still visible, with some higher noise frequency components managing to leak through the filter, resulting in a mottled appearance.

Ideal Low-Pass Filter

As an alternative to the average filter, a filter may designed explicitly with no sidelobes by forcing the frequency response to be zero outside of a given radial cutoff frequency Ω_c:

$$H(U,V) = \begin{cases} 1; & \sqrt{U^2 + V^2} \leq \Omega_c \\ 0; & \text{else} \end{cases} \tag{26}$$

or outside of a rectangle defined by cutoff frequencies along the U- and V-axes:

$$H(U,V) = \begin{cases} 1; & |U| \leq U_c \text{ and } |V| \leq V_c \\ 0; & \text{else} \end{cases}. \tag{27}$$

Such a filter is called *ideal low-pass filter* (ideal LPF) because of its idealized characteristic. We will study (27) rather than (26) because it is easier to describe the impulse response of the filter. If the region of frequencies passed by (26) is square, there is little practical difference in the two filters if $U_c = V_c = \gamma\Omega_c$.

The impulse response of the ideal low-pass filter (26) is given explicitly by:

$$h(m,n) = U_c V_c \, \text{sinc}(2\pi U_c m) \cdot \text{sinc}(2\pi V_c n). \tag{28}$$

where $\text{sinc}(x) = (\sin x / x)$. Despite the seemingly "ideal" nature of this filter, it has some major drawbacks. First, it cannot be implemented exactly as a linear convolution, because the impulse response (28) is infinite in extent (it never decays to zero). Therefore, it must be approximated. One way is to simply truncate the impulse response, which in image processing applications is often satisfactory. However, this has the effect of introducing *ripple* near the frequency discontinuity, producing unwanted noise leakage. The introduced ripple is a manifestation of the well-known *Gibbs phenomena* studied in standard signal processing texts [1]. The ripple can be reduced by using a tapered truncation of the impulse response [e.g., by multiplying (28) with a Hamming window] [1]. If the response is truncated to image size $M \times N$, the ripple will be restricted to the vicinity of the locus of cut-off frequencies, which may make little difference in the filter performance. Alternately, the ideal LPF can be approximated by a Butterworth filter or other ideal LPF approximating function. The Butterworth filter has frequency response [2]:

$$H(U,V) = \frac{1}{1 + \left(\sqrt{U^2 + V^2}/\Omega_c\right)^{2K}}. \tag{29}$$

and, in principle, can be made to agree with the ideal LPF with arbitrary precision by taking the filter order K large

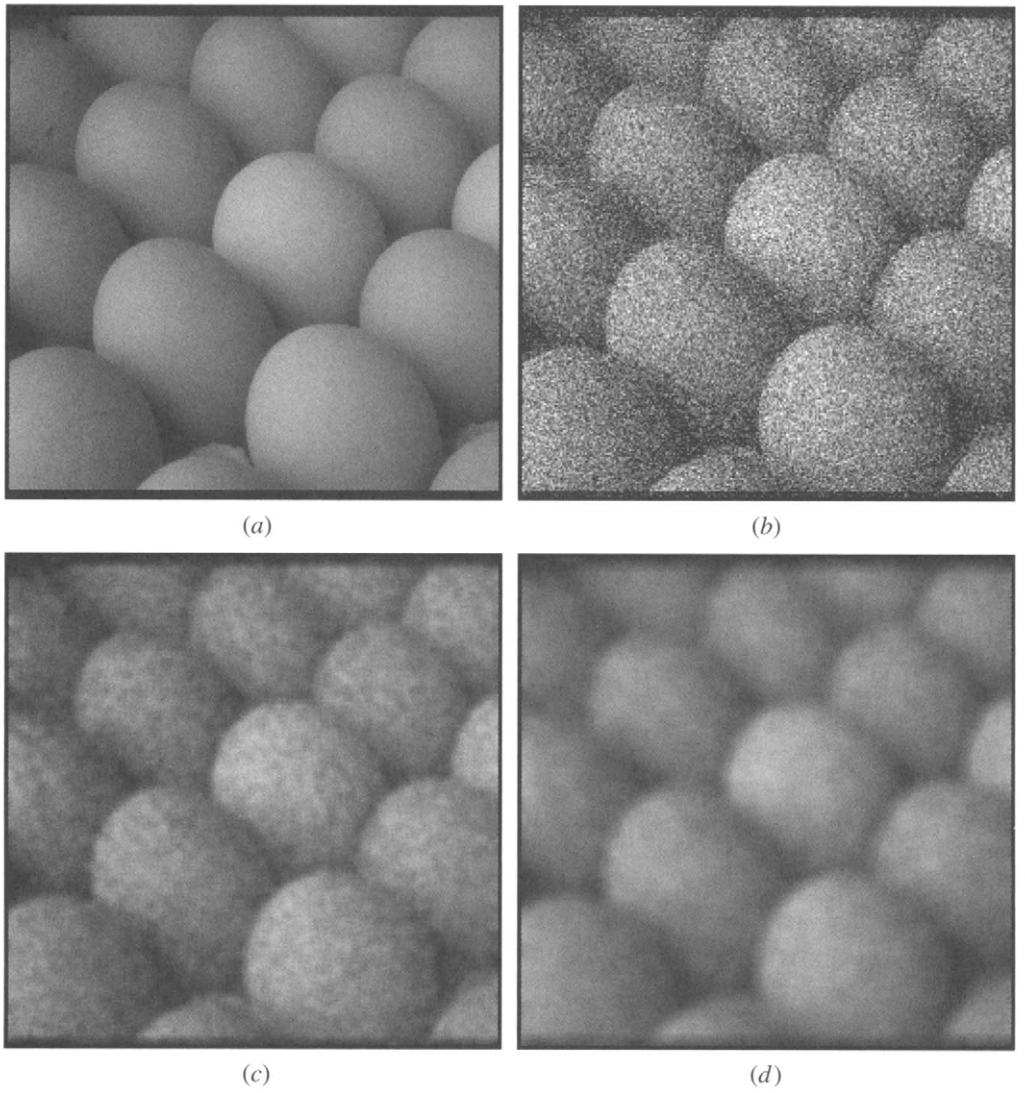


FIGURE 3 Example of application of moving average filter. (*a*) Original image "eggs"; (*b*) image with additive Gaussian white noise; moving average filtered image using (*c*) SQUARE(25) window (5 × 5) and (*d*) SQUARE (81) window (9 × 9).

enough. However, (29) also has an infinite-extent impulse response with no known closed-form solution. Hence, to be implemented it must also be spatially truncated (approximated), which reduces the approximation effectiveness of the filter [2].

It should be noted that if a filter impulse response is truncated, it should also be slightly modified by adding a constant level to each coefficient. The constant should be selected such that the filter coefficients sum to unity. This is commonly done because it is generally desirable that the response of the filter to the (0, 0) spatial frequency be unity, and since for any filter:

$$H(0,0) = \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} h(p,q). \tag{30}$$

The second major drawback of the ideal LPF is the phenomena known as *ringing*. This term arises from the characteristic response of the ideal LPF to highly concentrated bright spots in an image. Such spots are impulselike, and so the local response has the appearance of the impulse response of the filter. For the circularly symmetric ideal LPF in (26), the response consists of a blurred version of the impulse surrounded by sinclike spatial sidelobes, which have the appearances of rings surrounding the main lobe.

In practical application, the ringing phenomena create more of a problem because of the *edge response* of the ideal LPF. In the simplistic case, the image consists of a single one-dimensional step edge: $s(m,n) = s(n) = 1$ for $n \geq 0$ and $s(n) = 0$, otherwise. Figure 4 depicts the response of the ideal LPF with impulse response (28) to the step edge. The step

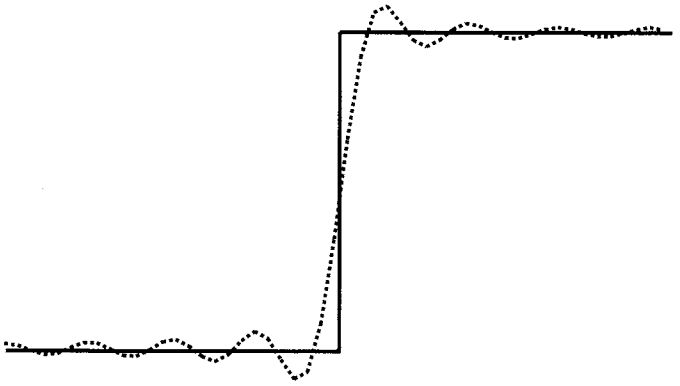


FIGURE 4 Depiction of edge ringing. The step edge is shown as a continuous curve, while the linear convolution response of the ideal LPF (28) is shown as a dotted curve.

response of the ideal LPF oscillates (rings) because the sinc function oscillates about the zero level. In the convolution sum, the impulse response alternately makes positive and negative contributions, creating overshoots and undershoots near the edge profile. Most digital images contain numerous steplike light-to-dark or dark-to-light image transitions; hence, application of the ideal LPF will tend to contribute considerable ringing artifacts to images. Because edges contain much of the significant information about the image, and because the eye tends to be sensitive to ringing artifacts, often the ideal LPF and its derivatives are not a good choice for image smoothing. However, if it is desired to strictly band-limit the image as closely as possible, then the ideal LPF is a necessary choice.

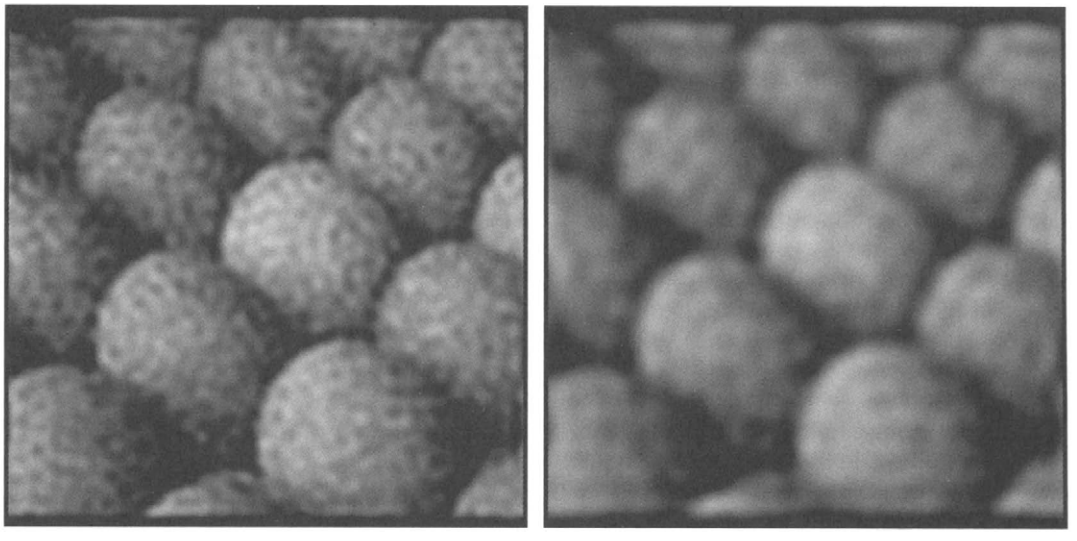
Once an impulse response for an approximation to the ideal LPF has been decided, the usual approach to implementation again entails zero-padding both the image and the impulse response, using the periodic extension, taking the product of their DFTs (using an FFT algorithm), and defining the result as the inverse DFT. This was done in the example of Fig. 5, which depicts application of the ideal LPF using two cutoff frequencies. This was implemented using a truncated ideal LPF without any special windowing. The dominant characteristic of the filtered images is the ringing, manifested as a strong mottling in both images. A very strong oriented ringing can be easily seen near the upper and lower borders of the image.

Gaussian Filter

As we have seen, *filter sidelobes* in the space or spatial frequency domains contribute a negative effect to the responses of noise-smoothing linear image enhancement filters. Frequency-domain sidelobes lead to noise leakage, and space-domain sidelobes lead to ringing artifacts. A filter with sidelobes in neither domain is the *Gaussian filter,* with impulse response

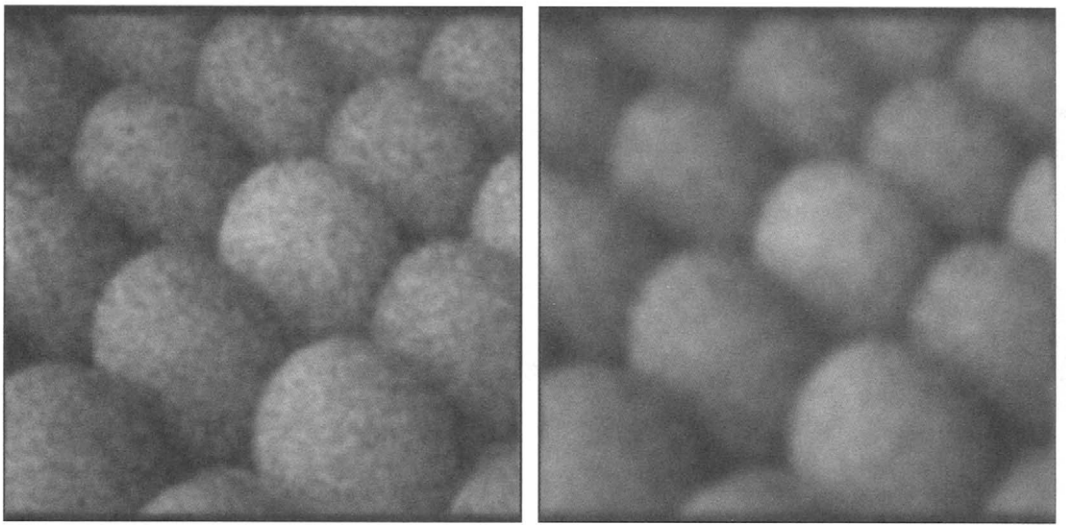
$$h(m,n) = \frac{1}{2\pi\sigma^2} e^{-(m^2+n^2)/2\sigma^2}. \tag{31}$$

The impulse response (31) is also infinite in extent, but falls off rapidly away from the origin. In this case, the frequency



(*a*) (*b*)

FIGURE 5 Example of application of ideal low-pass filter to noisy image in Fig. 3(*b*). Image is filtered using radial frequency cutoff of (*a*) 30.72 cycles/image and (*b*) 17.07 cycles/image. These cutoff frequencies are the same as the half-peak cutoff frequencies used in Fig. 3.



(a) (b)

FIGURE 6 Example of application of Gaussian filter to noisy image in Fig. 3(*b*). Image is filtered using radial frequency cutoff of (*a*) 30.72 cycles/image ($\sigma \approx 1.56$ pixels) and (*b*) 17.07 cycles/image ($\sigma \approx 2.80$ pixels). These cutoff frequencies are the same as the half-peak cutoff frequencies used in Figs. 3 and 5.

response is closely approximated by:

$$H(U,V) \approx e^{-2\pi^2\sigma^2(U^2+V^2)} \text{ for } |U|, |V| < \frac{1}{2} \tag{32}$$

which is also a Gaussian function. Neither (31) nor (32) shows any sidelobes; instead, both impulse and frequency response decay smoothly. The Gaussian filter is noted for the absence of ringing and noise leakage artifacts. The half-peak radial frequency bandwidth of (32) is easily found to be:

$$\Omega_c = \frac{1}{\pi\sigma}\sqrt{\ln\sqrt{2}} \approx \frac{0.187}{\sigma}. \tag{33}$$

If it is possible to decide an appropriate cutoff frequency $\gamma\Omega_c$, the cutoff frequency may be fixed by setting $\sigma = 0.187/\gamma\Omega_c$ pixels. The filter may then be implemented by truncating (31) using this value of σ, adjusting the coefficients to sum to one, zero-padding both impulse response and image (taking care to use the periodic extension of the impulse response implied by the DFT), multiplying DFTs, and taking the inverse DFT to be the result. The results obtained are much better than those computed using the ideal LPF, and slightly better than those obtained with the moving-average filter, because of the reduced noise leakage (Fig. 6).

Figure 7 shows the result of filtering an image with Gaussian filter of successively larger σ values. As the value of σ is increased, small-scale structures such as noise and details are reduced to a greater degree. The sequence of images shown in Fig. 7(*b*) is a *Gaussian scale–space*, where each scaled image is calculated by convolving the original image with a Gaussian filter of increasing σ value [3].

The Gaussian scale–space may be thought of as evolving over time t. At time t, the scale-space image $\mathbf{g}_t$ is given by

$$\mathbf{g}_t = \mathbf{h}_\sigma * f \tag{34}$$

where $\mathbf{h}_\sigma$ is a Gaussian filter with scale factor σ, and f is the initial image. The time-scale relationship is defined by $\sigma = \sqrt{t}$. As σ is increased, less significant image features and noise begin to disappear, leaving only large-scale image features.

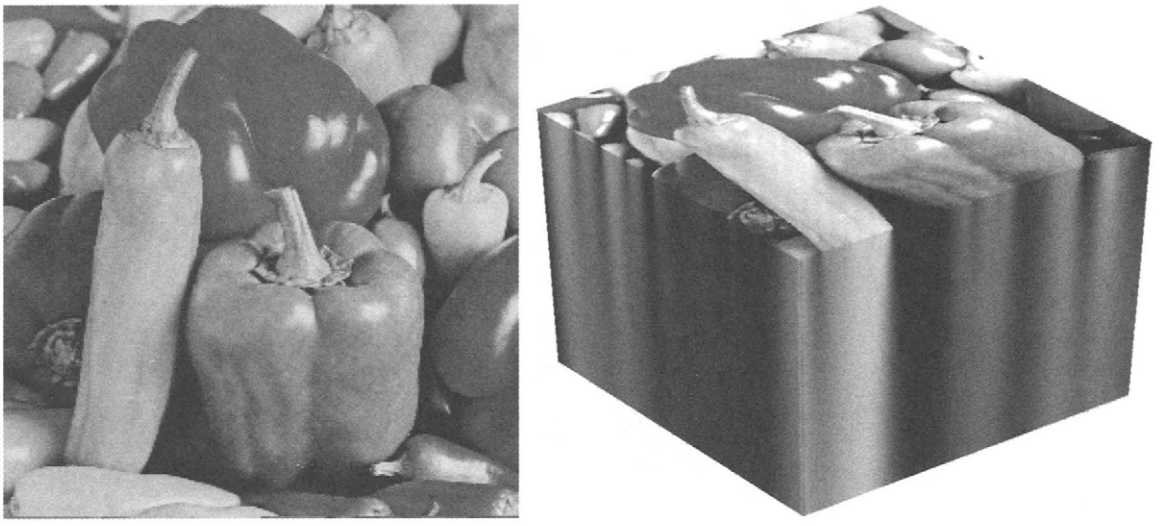
The Gaussian scale–space may also be viewed as the evolving solution of a partial differential equation [3, 4]:

$$\frac{\partial \mathbf{g}_t}{\partial t} = \nabla^2 \mathbf{g}_t \tag{35}$$

where $\nabla^2\mathbf{g}_t$ is the Laplacian of $\mathbf{g}_t$. For an extended discussion of scale-space and partial differential equation methods, see Chapters 4.15 through 4.18 of this *Handbook*.

4 Discussion

Linear filters are omnipresent in image and video processing. Firmly established in the theory of linear systems, linear filters are the basis of processing signals of arbitrary



(a) (b)

FIGURE 7 Depiction of scale-space property of Gaussian filter low-pass filter. In (*b*), the image in (*a*) is Gaussian-filtered with progressively larger values of σ (narrower bandwidths) producing successively smoother and more diffuse versions of the original. These are "stacked" to produce a data cube with the original image on top to produce the representation shown in (*b*).

dimensions. Since the advent of the FFT in the 1960s, the linear filter has also been an attractive device in terms of computational expense. However, it must be noted that linear filters are performance-limited for image-enhancement applications. From the several experiments performed in this chapter, it can be seen that the removal of broadband noise from most images via linear filtering is impossible without some degradation (blurring) of the image information content. This limitation is due to the fact that completely frequency separation between signal and broadband noise is rarely viable. Alternative solutions that remedy the deficiencies of linear filtering have been devised, resulting in a variety of powerful nonlinear image enhancement alternatives. These are discussed in Chapters 3.2 through 3.4 of this *Handbook*.

References

[1] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing* (Prentice Hall, Upper Saddle River, NJ, 1989).

[2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing* (Addison-Wesley, Boston, MA, 1993).

[3] A. P. Witkin, "Scale-space filtering," in *Proceedings of the International Joint Conference on Artificial Intelligence* (1983), 1019–1022.

[4] J. J. Koenderink, "The structure of images," *Biol. Cybern.* **50,** 363–370 (1984).