FUNDAMENTALS OF SPATIAL FILTERING

Spatial filtering is used in a broad spectrum of image processing applications, so a solid understanding of filtering principles is important. The name filter is borrowed from frequency domain processing where "filtering" refers to passing, modifying, or rejecting specified frequency components of an image. For example, a filter that passes low frequencies is called a lowpass filter. The net effect produced by a lowpass filter is to smooth an image by blurring it. We can accomplish similar smoothing directly on the image itself by using spatial filters.

Spatial filtering modifies an image by replacing the value of each pixel by a function of the values of the pixel and its neighbors. If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter. Otherwise, the filter is a nonlinear spatial filter. We will focus attention first on linear filters and then introduce some basic nonlinear filters.

LINEAR SPATIAL FILTERING

A linear spatial filter performs a sum-of-products operation between an image f and a *filter kernel*, w. The kernel is an array whose size defines the neighborhood of operation, and whose coefficients determine the nature of the filter. Other terms used to refer to a spatial filter kernel are *mask*, *template*, and *window*. We use the term *filter kernel* or simply *kernel*.

Figure 1 illustrates the mechanics of linear spatial filtering using a 3×3 kernel. At any point (x, y) in the image, the response, $g(x, y)$, of the filter is the sum of products of the kernel coefficients and the image pixels encompassed by the kernel:

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \cdots + w(0,0)f(x,y) + \cdots + w(1,1)f(x+1,y+1)$$

As coordinates x and y are varied, the center of the kernel moves from pixel to pixel, generating the filtered image, g, in the process. A filtered pixel value typically is assigned to a corresponding location in a new image created to hold the results of filtering.

Observe that the center coefficient of the kernel, $w(0,0)$, aligns with the pixel at location (x, y). For a kernel of size (m, n), we assume that $m = 2a + 1$ and $n = 2b + 1$, where a and b are nonnegative integers. This means that our focus is on kernels of odd size in both coordinate

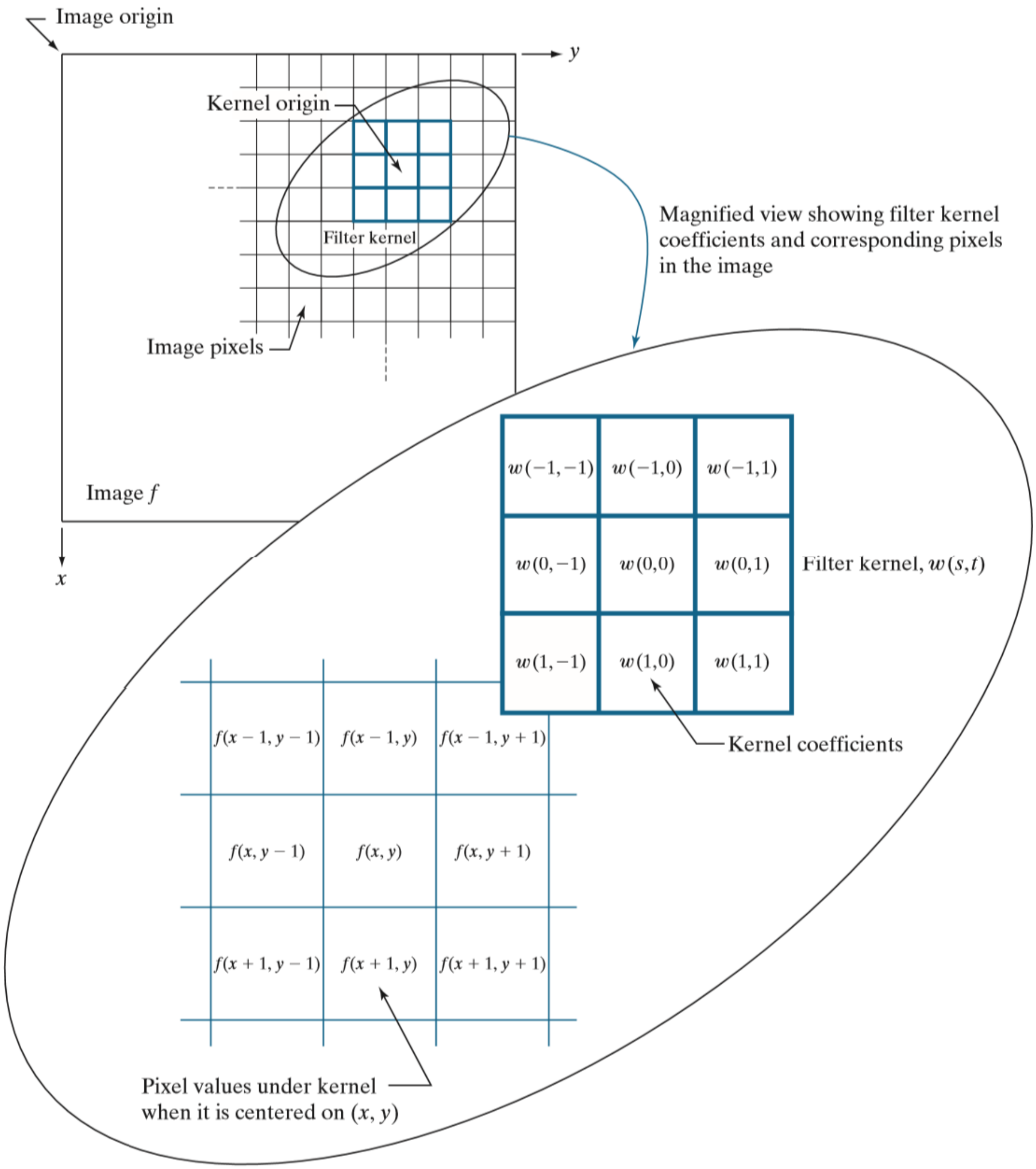

Figure 1: The mechanics of linear spatial filtering using a 3×3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.

directions. In general, linear spatial filtering of an image of size $M \times N$ with a kernel of size $m \times n$ is given by the expression

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

where x and y are varied so that the center (origin) of the kernel visits every pixel in f once.

SMOOTHING (LOWPASS) SPATIAL FILTERS

Smoothing (also called averaging) spatial filters are used to reduce sharp transitions in intensity. Because random noise typically consists of sharp transitions in intensity, an obvious application of smoothing is noise reduction. Smoothing is used to reduce irrelevant detail in an image, where "irrelevant" refers to pixel regions that are small with respect to the size of the filter kernel.

Linear spatial filtering consists of convolving an image with a filter kernel. Convolving a smoothing kernel with an image blurs the image, with the degree of blurring being determined by the size of the kernel and the values of its coefficients.

1. BOX FILTER KERNELS

The simplest, separable lowpass filter kernel is the box kernel, whose coefficients have the same value (typically 1). The name "box kernel" comes from a constant kernel resembling a box when viewed in 3-D. We show a 3×3 box filter in Fig. 2(a). An $m \times n$ box filter is an $m \times n$ array of 1's, with a normalizing constant in front, whose value is 1 divided by the sum of the values of the coefficients (i.e., $\frac{1}{\mathrm{mn}}$ when all the coefficients are 1's).

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

(a)

$\frac{1}{4.8976} \times$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

(b)

Figure 2: Examples of smoothing kernels: (a) is a box kernel; (b) is a Gaussian kernel.

Figure 3(a) shows a test pattern image of size 1024×1024 pixels. Figures 3(b)-(d) are the results obtained using box filters of size $m \times m$ with m = 3, 11, and 21, respectively. For m = 3, we note a slight overall blurring of the image, with the image features whose sizes are comparable to the size of the kernel being affected significantly more. Such features include the thinner lines in the image and the noise pixels contained in the boxes on the right side of the image. The filtered image also has a thin gray border, the result of zero-padding the image prior to filtering. The padding extends the boundaries of an image to avoid undefined operations when parts of a kernel lie outside the border of the image during filtering. When zero (black) padding is used, the net result of smoothing at or near the border is a dark gray border that arises from including black pixels in the averaging process. Using the 11×11 kernel resulted in more pronounced blurring throughout the image, including a more prominent dark border. The result with the 21×21 kernel shows significant blurring of all components of the image, including the loss of the characteristic shape of some components, including, for example, the small square on the top left and the small character on the bottom left. The dark border resulting from zero padding is proportionally thicker than before.

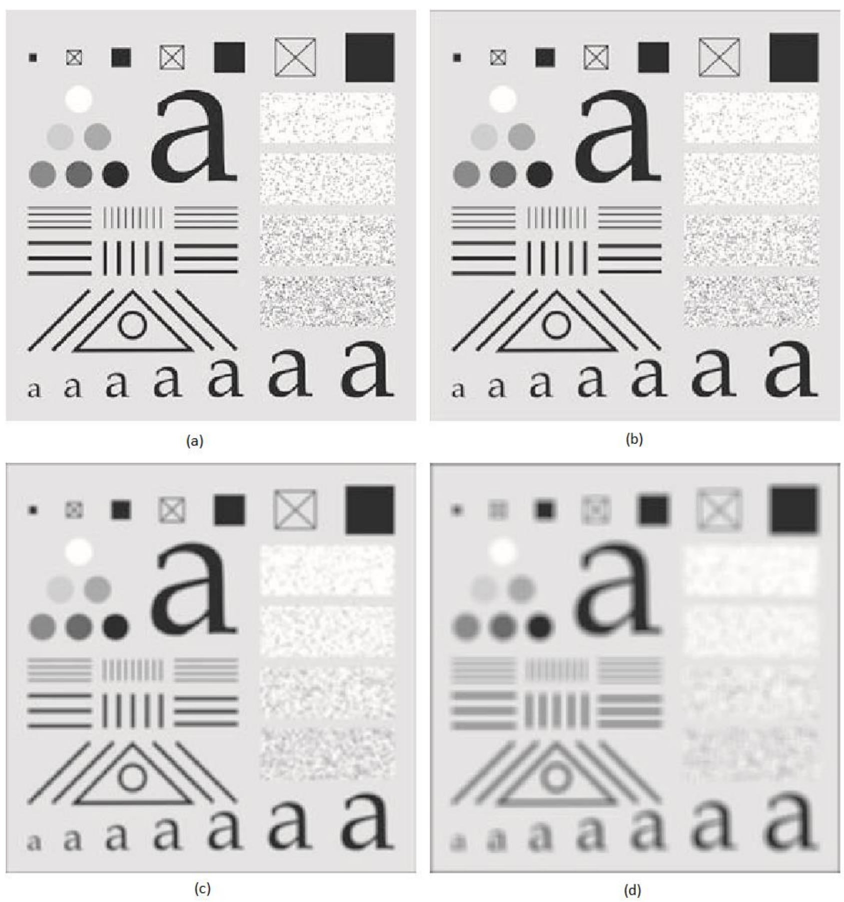

Figure 3: (a) Test pattern of size 1024×1024 *pixels. (b)-(d) Results of lowpass filtering with box kernels of sizes* 3×3 *,* 11×11*, and* 21×21 *, respectively.*

2. LOWPASS GAUSSIAN FILTER KERNELS

Because of their simplicity, box filters are suitable for quick experimentation and they often yield smoothing results that are visually acceptable. They are useful also when it is desired to reduce the effect of smoothing on edges. However, box filters have limitations that make them poor choices in many applications. For example, a defocused lens is often modeled as a lowpass filter, but box filters are poor approximations to the blurring characteristics of lenses. Another limitation is the fact that box filters favor blurring along perpendicular directions. In applications involving images with a high level of detail, or with strong geometrical components, the directionality of box filters often produces undesirable results.

The kernels of choice in applications such as those just mentioned are circularly symmetric (also called isotropic, meaning their response is independent of orientation). As it turns out, Gaussian kernels of the form

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$

To compare Gaussian and box kernel filtering, we repeat the example in the box kernel using a Gaussian kernel. Gaussian kernels have to be larger than box filters to achieve the same degree of blurring. This is because, whereas a box kernel assigns the same weight to all pixels, the values of Gaussian kernel coefficients (and hence their effect) decreases as a function of distance from the kernel center.

(a) (b) (c)

Figure 4: (a)A test pattern of size 1024×1024. (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21×21, with standard deviations $\sigma = 3.5$. (c) Result of using a kernel of size 43×43, with $\sigma = 7$. This result is comparable to Fig. 3(d). We used K = 1 in all cases.

The results in figures 3 and 4 showed little visual difference in blurring. Despite this, there are some subtle differences that are not apparent at first glance. For example, compare the large letter “a” in Figs. 3(d) and 4(c); the latter is much smoother around the edges. Figure 5 shows this type of different behavior between box and Gaussian kernels more clearly. The image of the rectangle was smoothed using a box and a Gaussian kernel with the sizes and parameters listed in the figure.

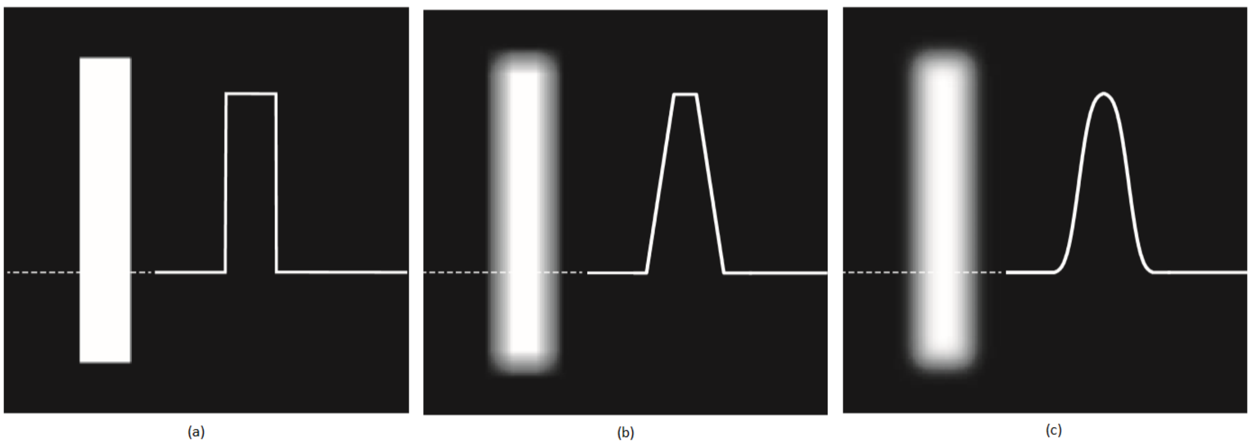

Figure 5: (a) Image of a white rectangle on a black background, and a horizontal intensity profile along the scan line shown dotted. (b) Result of smoothing this image with a box kernel of size 71×71, and corresponding intensity profile. (c) Result of smoothing the image using a Gaussian kernel of size 151×151, with $K = 1$ and $\sigma = 25$. Note the smoothness of the profile in (c) compared to (b).

References and further reading:

Digital Image Processing, 4th edition, Gonzalez, Rafael and Woods, Richard, 2018