INTENSITY TRANSFORMATIONS AND SPATIAL FILTERING

The term *spatial domain* refers to the image plane itself, and image processing methods in this category are based on direct manipulation of pixels in an image. This is in contrast to image processing in a transform domain which involves first transforming an image into the transform domain, doing the processing there, and obtaining the inverse transform to bring the results back into the spatial domain. Two principal categories of spatial processing are intensity transformations and spatial filtering. Intensity transformations operate on single pixels of an image for tasks such as contrast manipulation and image thresholding. Spatial filtering performs operations on the neighborhood of every pixel in an image. Examples of spatial filtering include image smoothing and sharpening.

THE BASICS OF INTENSITY TRANSFORMATIONS AND SPATIAL FILTERING

The spatial domain processes we discuss in this chapter are based on the expression

$$g(x,y) = T[f(x,y)] \tag{1}$$

where $f(x,y)$ is an input image, $g(x,y)$ is the output image, and T is an operator on f defined over a neighborhood of point (x,y). The operator can be applied to the pixels of a single image (our principal focus in this chapter) or to the pixels of a set of images, such as performing the elementwise sum of a sequence of images for noise reduction. Figure 1 shows the basic implementation of Eq. (1) on a single image. The point (x_0, y_0) shown is an arbitrary location in the image, and the small region shown is a *neighborhood* of (x_0, y_0). Typically, the neighborhood is rectangular, centered on (x_0, y_0), and much smaller in size than the image.

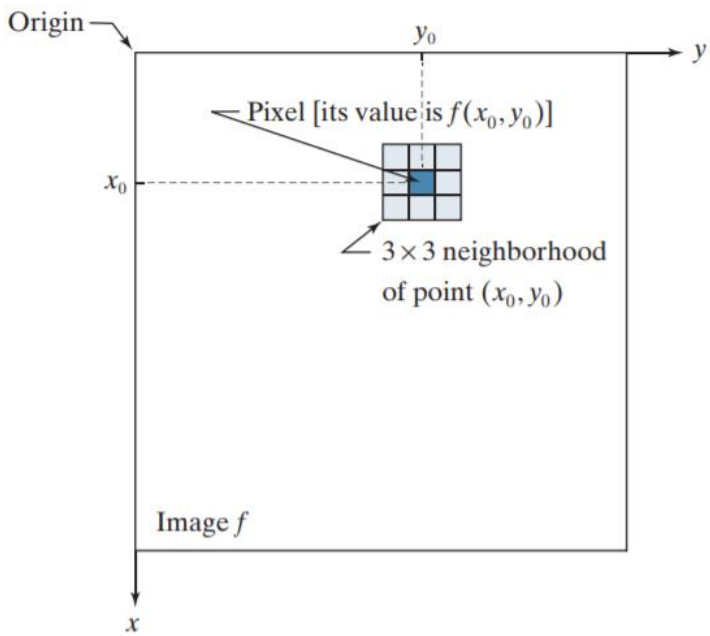

Figure 1: A 3×3 neighborhood about a point (x_0, y_0) in an image. The neighborhood is moved from pixel to pixel in the image to generate an output image.

The process that Figure 1 illustrates consists of moving the center of the neighborhood from pixel to pixel, and applying the operator T to the pixels in the neighborhood to yield an output value at that location. Thus, for any specific location (x_0, y_0), the value of the output image g at those coordinates is equal to the result of applying T to the neighborhood with origin at (x_0, y_0) in f. For example, suppose that the neighborhood is a square of size 3×3 and that operator T is defined as "compute the average intensity of the pixels in the neighborhood". Consider an arbitrary location in an image, say $(100, 150)$. The result at that location in the output image, $g(100{,}150)$, is the sum of $f(100{,}150)$ and its 8-neighbors, divided by 9. The center of the neighborhood is then moved to the next adjacent location and the procedure is repeated to generate the next value of the output image g. Typically, the process starts at the top left of the input image and proceeds pixel by pixel in a horizontal (vertical) scan, one row (column) at a time.

The smallest possible neighborhood is of size 1×1. In this case, g depends only on the value of f at a single point (x, y) and T in Eq. (1) becomes an *intensity* (also called a *gray-level*, or *mapping*) *transformation function* of the form

$$s = T(r) \tag{2}$$

where, for simplicity in notation, we use s and r to denote, respectively, the intensity of g and f at any point (x, y). For example, if $T(r)$ has the form in Fig. 2(a), the result of applying the transformation to every pixel in f to generate the corresponding pixels in g would be to produce an image of higher contrast than the original, by darkening the intensity levels below k and brightening the levels above k. In this technique, sometimes called *contrast stretching*, values of r lower than k reduce (darken) the values of s, toward black. The opposite is true for values of r higher than k. Observe how an intensity value r_0 is mapped to obtain the corresponding value s_0. In the limiting case shown in Fig. 2(b), $T(r)$ produces a two level (binary) image. A mapping of this form is called a *thresholding function*. Some fairly simple yet powerful processing approaches can be formulated with intensity transformation functions. Approaches whose results depend only on the intensity at a point sometimes are called *point processing* techniques, as opposed to the *neighborhood processing* techniques discussed in the previous paragraph.

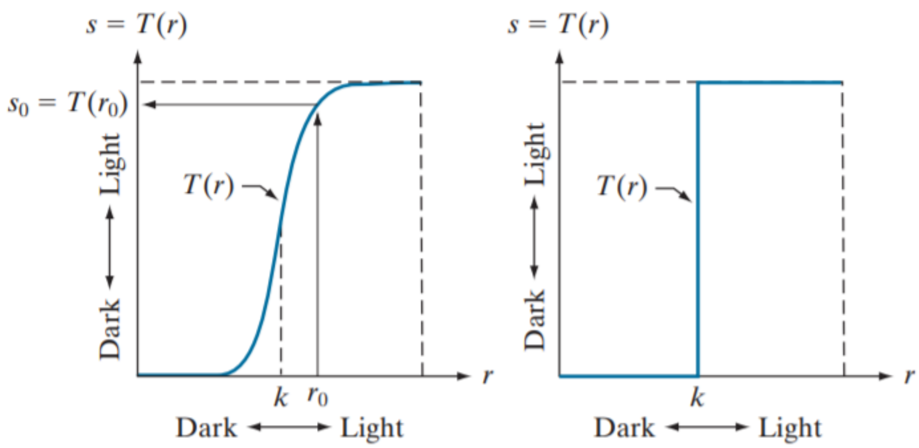

Figure 2: Intensity transformation functions. (a) Contrast stretching function. (b) Thresholding function.

Although intensity transformation and spatial filtering methods span a broad range of applications, most of the examples in this lecture are applications to image enhancement. Enhancement is the process of manipulating an image so that the result is more suitable than the original for a specific application. The word specific is important, because it establishes at the outset that enhancement techniques are problem-oriented. Thus, for example, a method that is quite useful for enhancing X-ray images may not be the best approach for enhancing infrared images.

IMAGE NEGATIVES

The negative of an image with intensity levels in the range $[0, L - 1]$ is obtained by:

$$s = L - 1 - r$$

This type of processing is used, for example, in enhancing white or gray detail embedded in dark regions of an image, especially when the black areas are dominant in size.

Reversing the intensity levels of a digital image in this manner produces the equivalent of a photographic negative. This type of processing is used, for example, in enhancing white or gray detail embedded in dark regions of an image, especially when the black areas are dominant in size. Figure 3 shows an example. The original image is a digital mammogram showing a small lesion. Despite the fact that the visual content is the same in both images, some viewers find it easier to analyze the fine details of the breast tissue using the negative image.

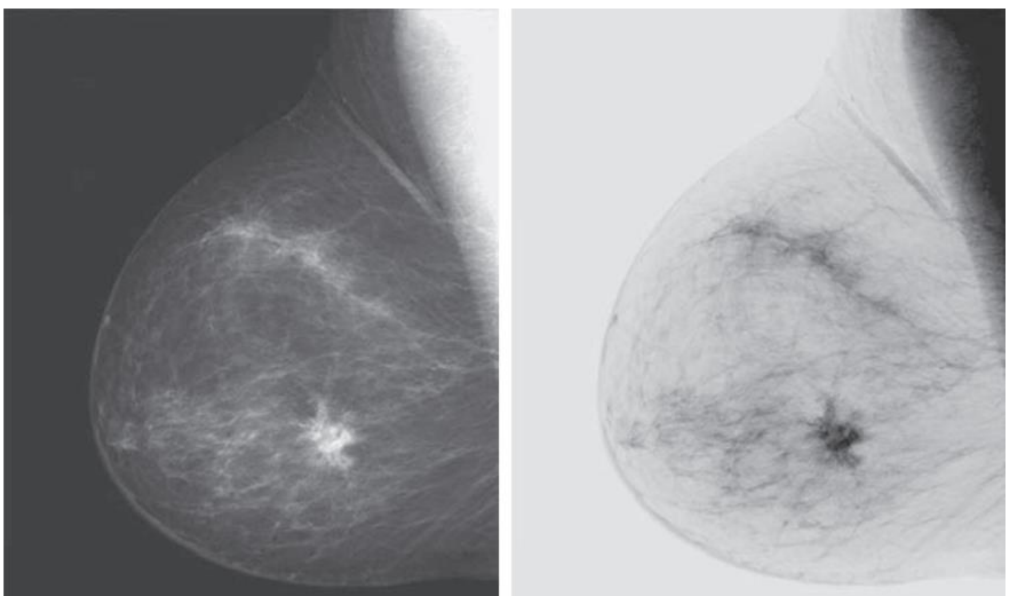

Figure 3: (a) A digital mammogram. (b) Negative image obtained using Eq. (3-3).

POWER-LAW (GAMMA) TRANSFORMATIONS

Power-law transformations have the form

$$s = cr^{\gamma} \tag{3}$$

where c and γ are positive constants.

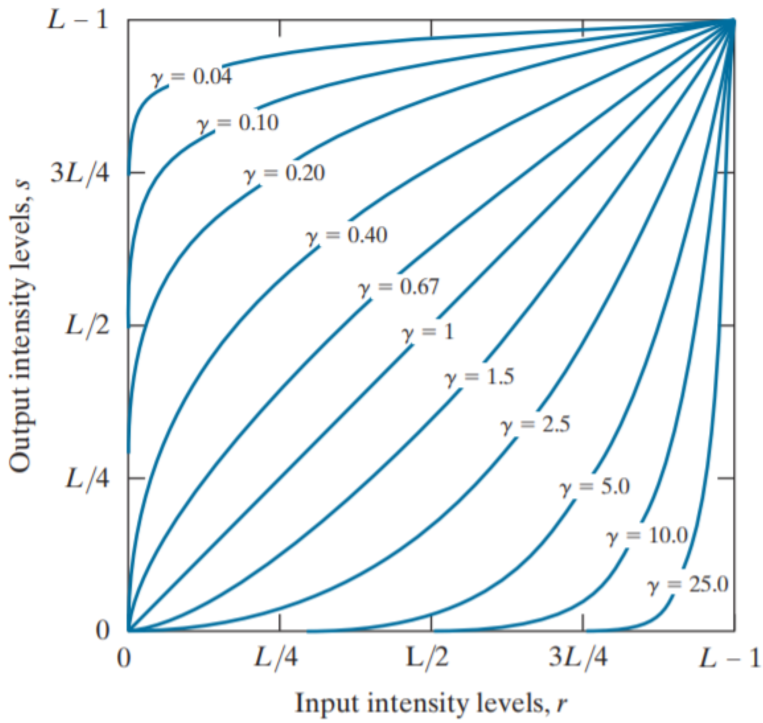

Figure 4: Plots of the gamma equation $s = cr^{\gamma}$ for various values of γ (c = 1 in all cases). Each curve was scaled independently so that all curves would fit in the same graph. Our interest here is on the shapes of the curves, not on their relative values.

INTENSITY-LEVEL SLICING

There are applications in which it is of interest to highlight a specific range of intensities in an image. Some of these applications include enhancing features in satellite imagery, such as masses of water, and enhancing flaws in X-ray images. The method, called intensity-level slicing, can be implemented in several ways, but most are variations of two basic themes. One approach is to display in one value (say, white) all the values in the range of interest and in another (say, black) all other intensities. This transformation, shown in Fig. 6(b), produces a binary image. The second approach, shown in Fig 6(c), brightens (or darkens) the desired range of intensities, but leaves all other intensity levels in the image unchanged.

Figure 5: (a) Aerial image. (b)–(d) Results of applying the transformation in eq. 3 with γ = 3.0, 4.0, and 5.0, respectively. (c = 1 in all cases.) (Original image courtesy of NASA.)

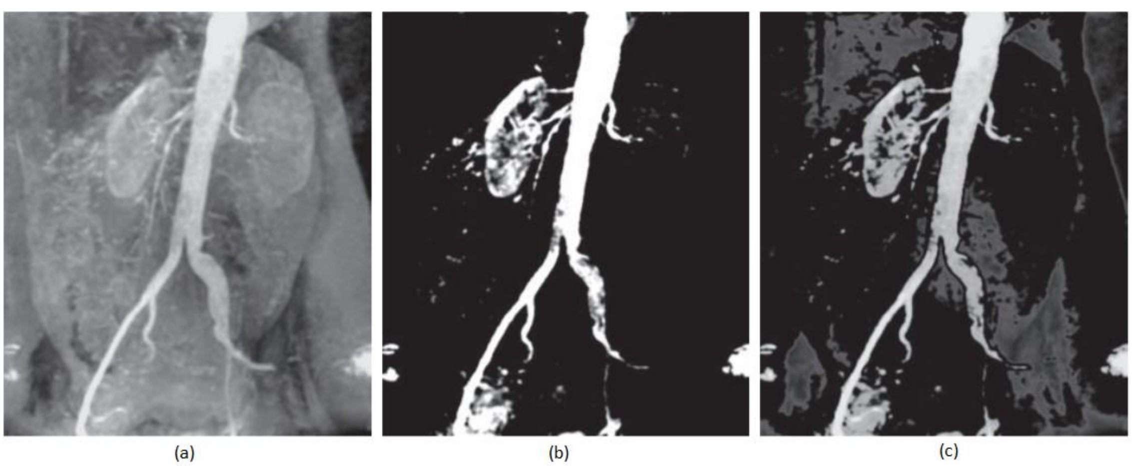

Figure 6: (a) Aortic angiogram. (b) Result of using a slicing transformation of the first type, with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation of the second type, with the selected range set near black, so that the grays in the area of the blood vessels and kidneys were preserved.

References and further reading:

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