Math 142 Reading Week 2

Abhi Uppal

September 13, 2021

1 Section 3.1: Point Operators

A **point operator** is the simplest kind of image transformation that takes an input pixel value and maps it independently of its neighbors. This is as opposed to **area-based** or **neighborhood** operators. Examples of point operators (or **point processes**) are brightness and contrast adjustments.

1.1 3.3.1: Pixel Transforms

A general image processing operator is a function that takes in one or more input images and produces an output image. This is denoted (in the continuous domain) as

$$g(\vec{x}) = h(f(\vec{x})) \text{ or } g(\vec{x}) = g(f_0(\vec{x}, \ldots, f_n(\vec{x}) \tag{1}$$

Where $\vec{x}$ is in the $D-$dimensional domain of the functions (usually $D = 2$ for images). For discrete images, the domain consists of a finite number of pixel locations, so we write

$$g(i, j) = h(f(i, j)) \tag{2}$$

Commonly we use multiplication and addition operators:

$$g(\vec{x}) = af(\vec{x}) + b \tag{3}$$

$a > 0$ and b are called the **gain** and **bias** parameters, respectively. These can also vary spatially (that is, $a = a(\vec{x})$ and $b = b(\vec{x})$).

Another commomly used **dyadic** (two-input) operator is the **linear blend** operator:

$$g(\vec{x}) = (1 - \alpha)f_0(\vec{x}) + \alpha f_1(\vec{x}). \tag{4}$$

This operator can be used to perform an image fade-in / out, like how we see in slade shows, by varying α from $0 \longrightarrow 1$.

1.2 3.1.2 Color Transforms

It helps to think of color images as highly correlated signals, rather than as arbitary vector-valued functions or collections of independent bands. Adding the same amount to each color increases the apparent intensity, hue, and saturation of the image.

Color balancing, for example, can be done by multiplying each channel by a different scaling factor. This can be written down in a $3x3$ matrix transformation.