

# Digital Image Processing - Gonzalez - Notes

An image may be defined as a two-dimensional function  $f(x, y)$  where  $x$  and  $y$  are spatial coordinates, and the amplitude of  $f$  at any pair of coordinates  $(x, y)$  is called the intensity or gray level of the image at that point.

The value at  $f$  at spatial coordinates  $(x, y)$  is a scalar quantity whose physical meaning is determined by the source of the image, and whose values are proportional to energy radiated by a physical source.

Digitizing (discretizing) the coordinate values is called *sampling*, digitizing the amplitude values is called *quantization*.

The coordinates of the image center are:  $(xc, yc) = (\text{floor}(M/2), \text{floor}(N/2))$

## Image Interpolation

Interpolation is the process of using known data to estimate values at unknown locations.

## Distance Measures

For pixels  $p, q$  and  $s$  with coordinates  $(x, y), (u, v), (w, z)$ , respectively,  $D$  is a **distance function** or metric if:

1.  $D(p, q) \geq 0$  ( $D(p, q) = 0$  iff  $p = q$ )
2.  $D(p, q) = D(q, p)$ , and
3.  $D(p, s) \leq D(p, q) + D(q, s)$

Some distance measures are the euclidean distance, the  $D_4$  distance (city-block), the chessboard distance, etc.

## Using image addition (averaging) for noise reduction

Suppose that  $g(x, y)$  is a corrupted image formed by the addition of noise,  $n(x, y)$  to a noiseless image  $f(x, y)$ ; that is:

$$g(x, y) = f(x, y) + n(x, y)$$

where the assumption is that at every pair of coordinates  $(x, y)$  the noise is uncorrelated and has zero average value. We assume also that the noise and image values are uncorrelated. The objective of the following procedure is to reduce the noise content of the output image by adding a set of noisy input images,  $\{g_i(x, y)\}$ .

If the noise satisfies the constraints just stated it can be shown that if an image  $\hat{g}(x, y)$  is formed by averaging  $K$  different noisy images,

$$\hat{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y)$$

then it follows that:

$$E\{\hat{g}(x, y)\} = f(x, y)$$

## On converting images to eight bits

Given a digital image  $g$ , an approach that guarantees that the full range of the values is “captured” into a fixed number of bits is as follows. First we perform operation  $g_m = g - \min(g)$ , which creates an image whose minimum value is 0, then we perform the operation

$$g_s = K \left[ \frac{g_m}{\max(g_m)} \right]$$

which creates a scaled image,  $g_s$ , whose values are in the range  $[0, K]$ . When working with 8-bit images, setting  $K = 255$  gives us a scaled image whose intensities span the full 8-bit scale from 0 to 255.

## Image Transforms

All image processing approaches discussed thus far operate directly on the pixels of an input image; that is, they work directly in the spatial domain. In some cases, image processing tasks are best formulated by transforming the input images, carrying the specified task in the **transform domain**, and applying the inverse transform to return to the spatial domain. A particularly important class of 2D linear transforms, denoted  $T(u, v)$ , can be expressed in the general form

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) r(x, y, u, v)$$

where  $f(x, y)$  is an input image,  $r(x, y, u, v)$  is called forward transformation kernel and is evaluated for  $u = 0, 1, 2, \dots, M-1$  and  $v = 0, 1, 2, \dots, N-1$ . As before,  $x$  and  $y$  are spatial variables. while  $M$  and  $N$  are the row and column dimensions of  $f$ . Variables  $u$  and  $v$  are called the transform variables.  $T(u, v)$  is called the forward transform of  $f(x, y)$ . Given  $T(u, v)$ , we can recover  $f(x, y)$  using the inverse transform of  $T(u, v)$ :

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v) s(x, y, u, v)$$

where  $s(x, y, u, v)$  is called the inverse transformation kernel.

### Some properties of the kernels

The forward transformation kernel is said to be separable if

$$r(x, y, u, v) = r_1(x, u) r_2(y, v)$$

In addition, the kernel is said to be symmetric if  $r_1(x, u)$  is functionally equal to  $r_2(y, u)$ , so that

$$r(x, y, u, v) = r_1(x, u) r_1(y, v)$$

## An introduction to the Discrete Fourier Transform

The nature of a transform is determined by its kernel, A transform of particular importance in digital image processing is the *Fourier transform*, which has the following forward and inverse kernels:

$$r(x, y, u, v) = e^{-j2\pi(ux/M+vy/N)}$$

and

$$s(x, y, u, v) = \frac{1}{MN} e^{-j2\pi(ux/M+vy/N)}$$

where  $j = \sqrt{-1}$ . Substituting these kernels into the general transform formulations gives the *discrete Fourier transform pair*. It can be shown that the Fourier kernels are separable and symmetric, and that separable and symmetric kernels allow 2-D transforms to be computed using 1-D transforms.

# Intensity Transformations and Spatial Filtering

“All the image processing techniques discussed in this chapter are implemented in the spatial domain, which we know is the plane containing the pixels of an image. Spatial domain techniques operate directly on the pixels of an image, as opposed, for example, to the frequency domain (Chapter 4) in which **operations are performed on the Fourier transform of an image, rather than on the image itself**”.

The spatial domain processes are based on the expression:

$$g(x, y) = T[f(x, y)]$$

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 process 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  $f$ .

## Some Basic Intensity Transformations

### Log Transformations

The general form of the log transformation is:

$$s = c \log(1 + r)$$

where  $c$  is a constant and it is assumed that  $r \geq 0$ .

In this transform, higher values of input levels are mapped to a narrower range in the output. One normally uses a transformation of this type to expand the values of dark pixels in an image, while compressing the higher-level values.

### Aliasing

Although aliasing is an inevitable fact of working with a sampled record of finite length, the effects of aliasing can be reduced by smoothing (lowpass filtering) the input function to attenuate its higher frequencies. This process, called anti-aliasing, has to be done before the function is sampled because aliasing is a sampling issue that cannot be undone after the fact using computational techniques.

## On the periodicity of the 1d and 2d fourier transform

The fourier transforms and its inverses are infinitely periodic in the  $u$  and  $v$  directions; that is,

$$F(u, v) = F(u + k_1 M, v) = F(u, v + k_2 N) = F(u + k_1 M, v + k_2 N)$$

and

$$f(x, y) = f(x + k_1 M, y) = f(x, y + k_2 N) = f(x + k_1 M, y + k_2 N)$$

where  $k_1$  and  $k_2$  are integers.

The periodicities of the transform and its inverse are important issues in the implementation of DFT based algorithm. The transform data in the interval from 0 to  $M - 1$  consists of two half periods meeting at point  $M/2$ , but with the lower part of the period appearing at higher frequencies.

*Attention:* For display and filtering purposes, it is more convenient to have in this interval a complete period of the transform in which the data are contiguous and ordered properly.

Centering the Fourier transform: the shifted array (2d) obtained by multiplying  $f(x, y)$  by  $(-1)^{x+y}$  before computing  $F(u, v)$ .

$$f(x, y)(-1)^{x+y} \iff F(u - M/2, v - N/2)$$

Using this equation shifts the data so that  $F(0, 0)$  is moved to the center of the frequency rectangle

## Fourier spectrum and phase angle

Because the 2D DFT is complex in general, it can be expressed in polar form:

$$F(u, v) = R(u, v) + jI(u, v) = |F(u, v)|e^{j\phi(u, v)}$$

where the magnitude:

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$$

is called the Fourier (or frequency) spectrum, and

$$\phi(u, v) = \arctan\left[\frac{I(u, v)}{R(u, v)}\right]$$

is the phase angle or phase spectrum. (The arctan must be computed using a four-quadrant arctangent function)

Finally the power spectrum is defined as

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$$

## On the wraparound error and padding

In a case where the closeness of the periods is such that they interfere with each other, causes what is commonly known as wraparound error. Fortunately, the solution to the wraparound error problem is simple. Consider two functions  $f(x)$  and  $h(x)$  composed of  $A$  and  $B$  samples, respectively. It can be shown that if we append zeros to both functions so that they have the same length, denoted by  $P$ , then the wraparound error is avoided by choosing:

$$P \geq A + B - 1$$

The padding of zeros could be appended also at the beginning of functions, or they could be divided between the beginning and end of the functions. It is simpler to append them at the end.

## Frequency Domain Filtering Fundamentals

Filtering in the frequency domain consists of modifying the Fourier transform of an image then computing the inverse transform to obtain the spatial domain representation of the processed result. This, given (a padded) digital image,  $f(x, y)$  of size  $P \times Q$  pixels, the basic filtering equation in which we are interested has the form:

$$g(x, y) = \text{Real}\{F^{-1}[H(u, v)F(u, v)]\}$$

NOTE: If  $H$  is real and symmetric and  $f$  is real, then the IDFT should yield real quantities in theory. In practice, the inverse often contains parasitic complex terms from roundoff error and other computational inaccuracies. Thus it is customary to take the real part of the IDFT to form  $g$ .

where  $F^{-1}$  is the IDFT,  $F(u, v)$  is the DFT of the input image,  $f(x, y)$ ;  $H(u, v)$  is a filter transfer function; and  $g(x, y)$  is the filtered output image. Functions  $F$ ,  $H$ , and  $g$  are arrays of size  $P \times Q$ , the same as the padded input image.