CONVOLUTION

Image convolution is a linear image processing operation where each dest pixel is computed based on a weighted sum of a set of (typically nearby) source pixels. For simplicity, label the pixels as a one-dimensional array. Let the n-th dest pixel have valueb_n, the p-th source pixel have value a_p, and the digital filter F have a set of nonzero values F_m for some set {m}, where typically the filter F_m is normalized so that Sum{m}F_m = 1. The filter works as follows:

b_n = Sum{m} F_m $a_{m - n}$ (convolution)

for each dest pixel n. The sum over the m terms in the convolution is the *inner loop* of the computation. The order of the indices on the RHS is a convention for the *convolution*, similar to the convention for the indices in the morphological erosion. Reversing the indices to take

c_n = Sum{m} F_m $a_{n - m}$ (correlation)

is called a *correlation*.

If the images are *N x M* pixels and the number of nonzero elements in the filter F is *s*, then the convolution requires *NMs* multiplications and additions. This grows linearly with the size of the convolution filter; for large filters and large images, the computation can be quite slow.

However, one often needs to use a large filter. For example, a low-pass filter is used to remove high frequencies in the image, prior to subsampling. (See the discussion of *anti-aliasing* in the section on image scaling.) The lower the passband of the filter, the larger the required filter.

There is one special situation where the computation does not depend on filter size; namely, where the filter is *rectangular* and *flat (i.e., with constant coefficients)*. We describe here a method that convolves an image with such rectangular flat filters of *arbitrary size*, and that performs the convolution in a time that is *independent of the size of the filter*! Essentially the same algorithm allows you to do fast median (and, more generally, rank-order) filtering of binary images. What these methods both have in common is the use of a pre-computed *accumulator array*. A similar algorithm using a one-dimensional accumulator of minimum or maximum values, rather than sums, can be used for fast grayscale morphological filtering with flat one-dimensional filters. Without the accumulator, grayscale morphological filtering requires either a scan at each location to find the min or max, or the maintenance of an ordered queue of pixel values as pixels are added and removed.

Source: http://www.leptonica.com/convolution.html