BINARY MORPHOLOGY

To distinguish itself from these, morphological image processing is sometimes called "image morphology" and "mathematical morphology," the latter perhaps to indicate the degree of abstractness that has been achieved. Image morphology was pioneered in France in the 1960s by Matheron and Serra, and further developed in Europe thereafter. But until the early 1990s, most image processing in the U.S. was linear: linear convolution and invertible transforms. Heavy mathematical machinery can be brought to bear on such linear processing, but there are many situations where non-linear image processing is required, especially in *image analysis* applications where decisions need to be made. (You can't make decisions with linear operations, which don't allow you to say "yes" or "no" -- only degrees of "maybe".) The neglect in the U.S. of these fundamental nonlinear operations reminds me of the old joke about someone who loses his keys at night but only searches for them under a lamp-post because "that's where the light is." I believe that the first doctoral thesis on image morphology in the U.S. was the 1985 work on nonlinear filters by Petros Maragos with Ron Schafer at Georgia Tech.

Around that time, several "machine vision" companies were founded in Michigan, largely to provide automated inspection for local auto manufacturers. Binary morphology was used as a principal technique because it is fast, memory efficient and provides general routines for pattern matching. These companies sold special purpose hardware because it was lucrative and it was difficult to put the algorithms on general purpose machines. Why was that? After all, the Motorola 68020 was available in 1985 and the Intel 386 was available in 1986. Each of these supported memory-protected multiprocessing with 32 bit addressing. The main problem was that DOS and Intel lived in a 16 bit world, using a 64K segmented memory model, and it was difficult to handle larger images. And the unix workstations that were built around the 68020 were relatively expensive. They were also slow -- these chips ran at a few MHz, several hundred times slower than the ubiquitous 3

GHz chips you can buy for $100 today. These machine vision systems were also popular because they were designed to be special purpose "turn-key" systems (though it was not always easy to turn the key, especially when you programmed them in an assembly-like language.) And so, into the early '90s, there was a market for this special purpose hardware. But the inexorable progress of the basic PC microprocessor, both in speed and in software tools for program development, has greatly reduced the demand for these special systems -- just as Silicon Graphics found that its business with special purpose graphics hardware was obliterated by cheap, fast Intel and AMD chips with gigaflop graphics performance. Today, processors are fast, memory is cheap, and we can use binary morphology for pattern matching at speeds that were not imagined even 10 years ago.

Binary morphology is about operations on sets. The sets are ON (black) pixels in a 2-dimensional image. As with all image processing operations, there is a *source* image that is operated on to produce a *destination* image. In the following, we use one set of conventions to describe the erosion and dilation operations as set operations. Other conventions exist, and you should keep this in mind when you read the literature.

The basic binary morphology operations are *dilation* and *erosion* . In a binary image, we refer to the foreground (black) pixels variously as "black", "foreground", "ON" or "1". We refer to the background (white) pixels variously as "white", "background", "OFF" or "0". Loosely speaking, dilation "smears" the foreground and erosion "thins" the foreground. These two operations are actually *dual* in that an erosion of the foreground is equivalent to a dilation of the background. Dilation can be implemented as follows: start with a cleared destination image (all OFF pixels). Then do a sequence of logical OR operations of the source image with the destination, each time with a specific shift, as determined by a pattern called a *structuring element* (*Sel*). The *Sel* is a 2-dimensional pattern of *hits*, all relative to an *origin* that is often referred to as the *center* of the *Sel*. It is a set as well. So there are really two sets involved in a morphological operation: the image and a *Sel*. For example, consider a *Sel* that is a horizontal pattern of 5 contiguous *hits* (x,y) = {(-2,0), (-1,0), (0,0), (1,0) and (2,0)}, with an origin at (0,0). Then a dilation of the image by this *Sel* involves initializing all dest pixels to OFF and then ORing the source five times with source shifts given by the five hits in the *Sel*.

An erosion can be implemented in a similar way. For example, you can initialize all dest pixels to ON and then do a set of logical ANDs between the dest and the shifted source. But there is one important difference: you take the shift to be *from the hit to the Sel origin*, rather than from the origin to the hit. You can think of the *Sel* as a set of vectors in two dimensions, and for erosion, you use an *inversion* of these vectors. Because of the AND, you end up with ON pixels in the destination only where the hits of the *Sel* can *all* fit on the ON pixels of the source image. In fact, the erosion can be implemented by placing the *Sel* with its origin at every ON pixel in the source image and, for every location where all hits in the *Sel* are placed on ON pixels in the image, an ON pixel is produced in the destination at the location of the *Sel* origin. Thus, the erosion is a *pattern matching* operation. The inversion of the *Sel* for erosion is required to make the erosion a dual of the dilation. The prescription for erosion just given is oversimplified; see the discussion in the next section on boundary conditions.

The inversion of the *Sel* for erosion is also required to make the *opening*, which is a sequence of erosion followed by dilation, both using the same *Sel*, have the property called *idempotence*. This means that if you do a second opening, there is no change after the first one. Operations that are idempotent have a special significance. The opening can be visualized as follows: *it gives you only those ON pixels where the Sel is able to fit entirely in the foreground*. (The erosion gave on pixels only at the *Sel* origin location of such a pattern match. Then by dilating those pixels, with the same *Sel*, you get *all* the pixels in the match.) So the opening *projects* out a subset of pixels of the source image, and a second opening gives the same result because the *Sel* fits in those pixels by construction. The *closing*, which is a dilation followed by an erosion with the same *Sel*, is also idempotent. It is the dual to the opening, because you can do a closing by opening the background. Regardless of the convention that is used to define erosion and dilation, the opening and closing operations have a unique definition. All four morphological operations have two important properties:

1. *Translational invariance*. The operation commutes with translation: if you translate the source and perform an operation, you get the same result as if you performed the operation first and then translated the result.

2. *Increasing*. If you have two images, one of which has a foreground that is entirely contained in the other, then the results of a morphological operation retain the same order of inclusion.

Unlike dilation and erosion, the result of opening or closing an image does not depend on the location of the origin of the *Sel*. (This is another reason for inverting the *Sel* for erosion.) A consequence of this is that opening and closing have special properties besides idempotence not shared by erosion and dilation. Namely, the opening is *anti-extensive*, which means that the set of ON pixels after opening is contained in the source image. Oppositely, the closing is *extensive* because the source image ON pixels are all contained in the image that results from closing it. The erosion and dilation do not have these properties because the result depends on the location of the *Sel* origin. For example, if you dilate an image with a *Sel* that has a single hit offset from the *Sel* origin, the result is simply a translation of the original image by the vector from the *Sel* origin to that hit. For this case, are the ON source pixels all contained in the ON pixels of the destination?

And there is more to morphology than translationally-invariant, increasing primitives

You might think that all important operations are translationally invariant, but this is not true. For example, any operation that scales the image size is not translationally invariant. For example, if an image is reduced by subsampling, the resulting image depends on the subsampling grid (i.e., on the specific pixels that are chosen to represent the reduced image). A translation of the image relative to this grid before subsampling can result in a different (translated) image from that produced by subsampling at the original grid location.

There is another important morphological operation, the *hit-miss* operation. It is a very general pattern match because it finds matches to parts of the image that have specified OFF pixels as well as ON pixels. For example, the lower edge of a horizontal line in the image can be found with a hit-miss *Sel* that has a row of *misses* below a row of *hits*. For a pixel to be set ON in the destination, all the hits must be on ON pixels *and* all the misses must be on OFF pixels in the source. (The operation is sometimes called ``hit-or-miss", but the operation is really a hit-*and*-miss.) Thus in its most general form, the *Sel* has in each location either a hit, a miss or a *don't-care*. The addition of misses to the *Sel* makes the hit-miss operation very useful, but it loses the increasing property. (Can you see why?). Consequently, it is not of great interest in set theory because without a preservation of image ordering, relatively few general statements can be made.

I should not leave you with the impression that this is all there is to binary morphology. There are many other image operations, such as thinning (with or without preservation of connectivity), that can be formed by sequences of basic operations. There are also many nonlinear transforms from binary to grayscale images, such as a simple distance transform that labels the minimum distance from each pixel to a pixel of opposite color. A more complicated nonlinear function labels *connected components* (a set of pixels of the same color for which each pixel in the set is adjacent, in either a 4- or 8-connected orientation, to another pixel in the set) by computing a measure of their "length" as the maximum, over all pairs of pixels in the set, of the minimum distance between the two pixels, taken over all paths between them that stay entirely within the set of pixels that constitute the connected component. Further, all the binary morphological operations can be generalized to grayscale, where the dilation is a *Max*operation and erosion is a *Min* operation. (Do you see how the binary dilation and erosion are a special case of taking the *Max* and *Min* operation? With a binary image, the max value is 1 and the min value is 0. An erosion places a value of 0 at the *Sel* origin unless all the pixels under the *Sel* are 1. Therefore, *the erosion selects the minimum of the binary pixel values*.) And once you go to grayscale, the number of interesting nonlinear operations explodes. For example, much work has been done on image segmentation, typically using seeds and region growing.

Source : http://www.leptonica.com/binary-morphology.html