A System for Fast Erosion and Dilation of Bi-level Images

J.R. Parker
Department of Computer Science
University of Calgary, Alberta

Abstract

Computer image processing techniques are often used in, for example, the analysis of thin sections of reservoir rock because of the large amounts of data contained in a single digitized section image. Erosion and dilation are operations frequently used in this type of work to iteratively smooth the pore perimeters and in estimating pore radii, volume, and roughness. Because of the size of each image, erosion and dilation of pore complex images is a time consuming process. A recent method called *global* erosion is much faster whether used on small, in-memory images or large images residing on a file. Use of this method should allow processing of larger images, or a greater number of small images, than do the standard methods

Keywords: Image processing, mathematical morphology, erosion/ dilation, Minkowski operators, medial axis transform.

Morphological analysis is commonly performed on certain kinds of digital picture. As an example, most observations of pores in reservoir rocks are obtained from thin sections or SEM images. Another example is an image containing biological cells, where morphology can be important for classification purposes Smeulders & ten Kate 1987,Young et al 1981), and another is medical imaging (for example, Silage & Gil 1988). These images can be too large to properly analyze by hand, being up to 4096 by 4096 data points in size, and computer processing of the images is usual. Generally the microscope image is scanned by a video digitizer, and the data is then transmitted to a computer, often a microcomputer. The data is in the form of rows and columns of intensity values, called *grey levels*; in the simplest form this is a collection of zero (0) values (for black, pore areas) and one (1) values (for white, non-pore areas). This 0-1 data can be compacted for archival storage by packing eight of these values into one byte. The goal is to obtain information concerning the size, shape, and roughness of the objects (pores in the rock), and this is done using computer software, sometimes with the assistance of special purpose image processing devices.

Basic to much of the analysis of pore complex images is the concept of image *erosion* and *dilation*. (Matheron 1967,1975, Serra 1982), also called shrinking and expanding by some (Sternberg 1982). Erosion involves the removal of layers of each pore, like peeling an onion. This is done by first noting which data elements in each pore are on the pore boundary, then removing those data elements by setting them to 1. This has the general effect of smoothing out irregularities in the pore perimeter, and can be repeated many times. For example, consider the small image:

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

When this is eroded by one layer, the result is:

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Layers could then be added back to the outer edges of the eroded pores, restoring the pores to an approximation of their original size and shape but without the surface irregularities; this is *dilation*, and applied to the image above gives:

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Note that the general size and shape of the dilated image is the same as the original uneroded image, except that the 'bump' on the original image has been removed (smoothed). .P The above represents erosion/dilation (E-D) in its simplest, although commonly applied, form. E-D is related to *opening* and *closing* and to the *medial axis transform* [Calabi & Hartnett 1968, Marshall 1989,Pratt 1978) in that all are *generalized Minkowski* operators. In its most general form a dilation can be expressed as a Minkowski sum of two sets I (image) and S (structuring element):

.EQ I~+~S~=~ union from {p epsilon~ I} {~S sub p}, ~~~{S sub p}~=~left {s+p|s epsilon S right } .EN

The set S can be arbitrarily complicated but is most often a disk or a small region. For example, the case in which S is a 3x3 region corresponds to the 'onion' analogy above. Similarly, a general erosion can be expressed as a Minkowski difference:

.EQ I~-~S ~=~ inter from {s epsilon~ S} {~I sub s} .EN

Erosions and dilations are frequently carried out in pairs. A dilation of I by S followed by an erosion of the result by S is called a *closing*. The result is an image in which the gaps and holes in I that are smaller than S in size are filled in (closed). An *opening* is an erosion of I by S followed by a dilation of the result by S, and results in the narrow regions and boundary irregularities of I being removed.
While this is is a general description of the terms *erosion*, *dilation,* *opening*, and *closing*, the most common situations have the set S as a 3x3 'mask':

```
1 1 1
1 1 1
1 1 1
```

In this case an erosion strips a layer of pixels from all of the regions in I, and a dilation adds a layer. An opening of size N involves N erosions followed by N dilations, and a closing of size N is therefore N dilations followed by N erosions. These definitions will be understood for the remainder of this discussion.

Erosion-dilation can be used to determine size and roughness parameters of the pores in the reservoir rock sample (Ehrlich et al 1984,Rink 1976). For example, average pore size can be found simply by counting the number of erosions needed to remove the pore altogether. .i Roughness .r can be found by opening and then finding the differences between the resulting image and the original. Depending on the number of layers removed and restored, irregularities of known sizes will remain in the difference image and can then be counted. .P Erosion-dilation is usually performed in cycles (Rosenfeld & Kak 1982) corresponding to openings and closings. For example, an image could have one layer eroded and then would be dilated by one layer again (opening of 1). After data obtained from this cycle had been computed, the image would be eroded by two layers and dilated by two layers (opening of 2) to collect more data. This process can be repeated until the erosion process removes all pixels belonging to a given pore, at which point the pore cannot be restored by dilation (.i ultimate .r erosion (Serra 1982) .r). At least one pixel must normally remain to provide a 'seed' for the dilation process. .P For small images, say up to 512 by 512 picture elements (.i pixels .r) the entire image can be kept in a microcomputer memory at one time. Each erosion step requires that the entire image be examined at least once (sometimes twice, depending of the method), and a dilation step also needs a scan of all of the data. Since a typical pore image can be eroded by about thirteen layers before all pores are removed, all erosions and dilations of an image requires referencing about 48 million data elements. If sufficient computer memory space is available the time needed for erosions can be shortened. The eroded image can be saved in a second image, dilated, then the saved image can be eroded by one more layer. .(z F .sp 1 .ce 1 .b Insert Figure 1 Here .r .sp 1 .)z .sp 1 .b .ce 1 1. Example of Erosion and Dilation .r .sp 1 .P Figure 1a shows a small simulated pore-complex image which is to be eroded. Assume that all pixels in the image have a value of 0 or 255, where a 255 value represents a pixel that belongs to a pore and a 0 value represents rock; the 255 values are represented by '*' characters in the figure. A standard erosion method can be described as follows: .br .i Algorithm .b ERODE: .r .br .sz 10 .ip 1) Look for a pixel with a value of 255. .ip 2) If one of the eight immediate neighbors of this pixel has a value of zero, then .i mark .r this pixel. This is done by changing its value to, say, 128. .ip 3) Continue from step 1 until all pixels have been examined. .ip 4) Look at the image again, this time searching for a .i marked .r pixel. .ip 5) Change the value of this marked pixel to zero, thereby removing it. .ip 6) Continue from step 4 until all .i marked .r pixels have been set to zero. .lp .sz 12 .P Figures 1b and 1c represent erosions of Figure 1a by two pixels/layers and four pixels/layers respectively using this method. The value of 255 represents pure white by convention, and allows the pixel values to be stored in a single memory byte. The standard method used for computing in-place dilations is similar. Figures 1d and 1e show the result of dilating Figures 1b and 1c, by two and four pixels respectively. .P The fast erosion method is based on the construction of a distance map of the pore image, where the numerical value of each pixel belonging to a pore is replaced by a new value representing the distance (number of pixels) of that pixel from the boundary of the pore. Pixels on the pore boundary would be given a value of 1, since they are 1 pixel away from the boundary, pixels that are 2 pixels away from the boundary would be given the value 2, and so on. The result has the appearance of a contour map, where the contours represent the distance from the edge of the pore. .(z F .sp 1 .ce 1 .b Insert Figure 2 Here .r .sp 1 .ce 1 Illustration of 'Global' Erosion. .sp 1 .)z .P It is clear that an erosion of one layer of pixels from the image will remove (set to 0) those pixels at a distance of one from the boundary, and erosion of two layers will remove those pixels at a distance of .i two or less .r from the boundary. The distance map image contains enough information to perform an erosion by any number of pixels in just one pass through the image; in other words, all possible erosions have been encoded into one picture. This .i globally eroded .r image can be produced in just two passes through the original image. The method is (Parker 1988): .br .i Algorithm .b GERODE .r .br .sz 10 .ip 1) Starting at the upper left of the image, scan along successive rows, from left to right, until a pixel with a value of 255 is found. This becomes the current pixel. .ip 2) Look at all 8 possible neighbors of the current pixel, and let V be the .i minimum .r value found in these pixels. .ip 3) Set the value of the current pixel to V+1. .ip 4) Continue from step 1 until all pixels have been examined. .ip 5) Now starting at the lower right of the image, scan along successive rows from right to left moving UP the image, looking for a pixel whose value is non-zero. This becomes the current pixel. .ip 6) Look at all 8 neighbors of the current pixel, and let V be the minimum value found in these pixels. .ip 7) If V has a value less than that of the current pixel, set the value of the current pixel to V+1. .ip 8) Repeat from step 5 until all pixels have been examined. .lp .sz 12 .P Steps 1 through 4 represent the first phase of the global erosion, and steps 5 through 8 represent the second phase. After the first phase is complete the pixel values represent the minimum distance, expressed as numbers of pixels, to the nearest upper, upper left, left, and upper right boundaries. The second phase must be performed by scanning in the opposite direction from the first in order to collect the distances in the other four directions in one pass. .P The resulting image has all possible erosions encoded as the pixel values, but it is important to note that no erosion yet taken place. It would normally not be necessary to actually perform an erosion by removing pixels, since the image will be immediately dilated again. If the actual eroded image is desired a simple thresholding operation will do the job. Figure 2a shows the result of globally eroding the pore image of Figure 1a. To remove N layers of pixels from a globally eroded image, simply set all pixels that have a value less than of equal to N to zero, and set all other pixels to 255; this can be done in one pass through the data. .P The process of .i fast dilation .r must be applied only to an image that has been globally eroded, since the procedure depends on the existence of the distance map, and results in an image that has been opened by N pixels. The method is: .br .i Algorithm .b FDILATE: .r .br .ip 1) Scan the image for a pixel with a value greater than zero that is also .i unmarked. .r Let this pixel be the current pixel. .ip 2) If the value of the current pixel is less than $k+1$ then set the current pixel to zero; if the value of the current pixel is greater than $k+1$ then .i mark .r the current pixel. In either case, skip to step 4. If the current pixel has a value equal to $k+1$ go to step 3. .ip 3) The current pixel must have a value of $k+1$. .i Mark .r the current pixel, and also .i mark .r all pixels within k pixels of the current pixel. .ip 4) Repeat from step 1 until all pixels have been examined. .lp .sz 12 .P The marked pixels in the image that results belong to the dilated pore; all other pixels are background. Figures 2b and 2c show the result of globally dilating the image of Figure 2a by 2 and 4 pixels respectively. Notice that there is no difference between these dilated images and those of Figures 1d and 1f. The blanked areas of Figure 2b and 2c show where the opening has left a gap in the image. .P If further dilations are needed, the original globally eroded image may be saved and dilated by a different number of pixels or the globally dilated image can be unmarked and re-dilated. Depending upon the method chosen for marking pixels, the globally dilated image may need to be thresholded before it is displayed. If a closing is desired it can be accomplished using a slight variation of the FDILATE process. .sp 1 .b .ce 1 3. Large Images .r .P It is not unusual to deal with images that are 4096 rows by 4096 columns in size. These are generally bi-level images obtained from a digitizer, and are packed into bytes at the rate of 8 pixels per byte (one bit per pixel). Global erosion and dilation can be performed on these images at the cost of expanding the images to one byte per pixel, but we need only four rows of storage for the image in memory. Two passes through the image file are needed to perform a global erosion, and then two passes for each dilation. .P Global erosion of the image requires that we have three rows of the image at one time, since all eight neighboring pixels must be available according to step 2 of the GERODE algorithm above. The center row would be the one being operated on; when that row is completed it is written out, and a new row is read in to replace the row before it. This scrolling though the file continues until the last row is seen, which ends the first pass. The second pass involves reading the file just written, but in reverse order, and scanning each row in reverse order too, to simulate the bottom to top, right to left scan needed in the second pass. When this is done, copying to a second file, the global erosion is complete. .P This is a small modification to the original GERODE method. Most of the modifications have the function of performing file manipulations, such as reading in a row, or buffer manipulations involved in scrolling through the image file. The process of dilation, on the other hand, requires a good deal more thought. .P A dilation of K pixels can be done as K dilations of one pixel, but this is quite costly. Each dilation requires a pass through a very large file, and input/output is the most time consuming operation on most computers. It is possible to dilate a globally eroded image by any number of layers in just two passes, and using four rows of image in memory. The basic plan is to examine each row as it is read in to see which pixels will be the 'seed' of the dilation. According to the algorithm FDILATE above, step 2, these seed pixels will have a value of K+1 or more, where K is the number of layers to be dilated. The global dilation method would set all pixels within K pixels of each seed, but since we have only one row of the image this cannot be done. .P What is possible is to remember which pixels to mark on the next rows of the image as it is read in. For each seed pixel found, the next K pixels to the left and right and below it will be marked. The K pixels on the left and right diagonals below will also need to be marked. For example, consider the seed pixel in Figure 3a. To dilate by 2 pixels, simple set (or mark) all pixels within 2 pixels of the seed, as seen in Figure 3b. If only one row at a time can be kept in memory, then the best that we can do after one pass though the image is seen in Figure 3c. Rows previous to the only on which the seed resides cannot have been affected by the presence of the seed. Another pass, from end of file to beginning, is needed to finish the dilation. .(z F .sp 5 .ce 1 Figure 3 goes here .sp 3 .ce 1 Figure 3 - Sequential Dilation of an Image .r .sp 1 .)z .P The information concerning the presence of seed pixels on previous rows must be kept for at least the next K rows on both passes. This is done by using three more rows of storage, which will be called image(2), image(3), and image(4). Image(2) remembers the presence of a seed pixel in the same column, but some rows above. Image(3) keeps track of seeds on a left-to-right diagonal some rows above, and image(4) keeps track of seeds on a right-to-left diagonal above. Each time a new row is read in from the file, all pixels that reside in columns corresponding to non-zero entries in any of these three arrays is marked. Then image(3) and image(4) are shifted right and left respectively, and all non-zero values in all three arrays are decremented. Finally, the current row of the image is scanned for seed pixels, and when one is found we set the corresponding entry in each of the three arrays to be K, the number of layers of dilation. As well, the K columns adjacent to each set pixel are marked on this row. .P This procedure is not quite complete. Consider the row: .sz 8 .ls 1 .nf

0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0
.sz 12 .ls 2 .fi Assume that the three arrays are all zero at this point. The method above specifies that the the columns that correspond to the 3 in the row will be set to K. In this case, let K=2. Then the three data arrays will be: .sz 8 .ls 1 .nf
Image(2):
0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0
Image(3): 0 0 0 0 0 0 0 0
0 0 2 0 0 0 0 0 0 0 0 0 Image(4):
0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0
.sz 12 .ls 2 .fi Note that Image(3) and Image(4) have their values shifted by one column in the direction of their diagonal. Now the columns in the current row are marked, and the current row looks like: .sz 8 .ls 1 .nf

0 0 0 0 0 0 0 * * 3 * * 0 0 0 0 0 0 0 0
.sz 12 .ls 2 .fi This row is written out to a file, and another one is read in. Assume that the next row has no seed pixels, for simplicity. The procedure above will mark the pixels in the current row that correspond to non-zero entries in each of image(2), image(3), and image(4), resulting in: .sz 8 .ls 1 .nf

0 0 0 0 0 0 0 0 * * * 0 0 0 0 0 0 0 0 0
.sz 12 .ls 2 .fi Now all elements in the three arrays (not the image row) are decrement and shifted, resulting in: .sz 8 .ls 1 .nf
Image(2):
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
Image(3): 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 Image(4):
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
.sz 12 .ls 2 .fi After this process is repeated once more, all arrays will be zero. The result, in terms of marked pixels in the image, will be: .sz 8 .ls 1 .nf

0 0 0 0 0 0 0 * * 3 * * 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 * * * 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 * 0 * 0 * 0 0 0 0 0 0 0 0
.sz 12 .ls 2 .fi This is not quite correct, and the reason is that the filled square 5 by 5 region contains more than just vertical and diagonal elements. The missing pixels can very easily be filled by two very simple additions to the method. After the arrays image(3) and image(4) are computed, their values should be compared column by column to image(2). If either image(3) or image(4) has a value greater than that found in the corresponding column of image(2), then image(2) is set to that greater value. This modification alone results in the dilation above now appearing as: .sz 8 .ls 1 .nf

0 0 0 0 0 0 0 * * 3 * * 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 * * * 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 * * * * * 0 0 0 0 0 0 0 0
.sz 12 .ls 2 .fi The second modification involves setting K columns around the seed column in image(3) and image(4) to K-J+1, where J is the distance, in columns, between the seed column and the column involved. Hence, the first step in the dilation is: .sz 8 .ls 1 .nf

Row I:
0 0 0 0 0 0 0 * * 3 * * 0 0 0 0 0 0 0 0
Image(2):
0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0
Image(3): 0 0 0 0 0 0 0 0 0 0 2 1 0 0 0 0 0 0 0 0 0 Image(4):
0 0 0 0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0
.sz 12 .ls 2 .fi Now image(2) columns are set to the maximum of image(2), image(3), or image(4). The next step is reading the next row and marking pixels in this, initially empty, row if they correspond to non-zero entries in image(2), image(4). The result is: .sz 8 .ls 1 .nf

Row I+1:
0 0 0 0 0 0 0 * * * * * 0 0 0 0 0 0 0 0
Image(2):
0 0 0 0 0 0 0 1 2 2 2 1 0 0 0 0 0 0 0 0
Image(3): 0 0 0 0 0 0 0 0 0 0 2 1 0 0 0 0 0 0 0 0 0 Image(4):
0 0 0 0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0
.sz 12 .ls 2 .fi Now all entries in the three arrays are decremented, shifted where appropriate, and then image(2) is again set to the maximum of image(2), image(3), and image(4). This results in: .sz 8 .ls 1 .nf

Row I+1:
0 0 0 0 0 0 0 * * * * * 0 0 0 0 0 0 0 0
Image(2):
0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0
Image(3): 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 Image(4):
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
.sz 12 .ls 2 .fi Finally, the next image row (number I+2) is marked, and the final decrement of the three arrays results in their contents being reduced to all zeros. When all rows of the image have been scanned in the forward direction, the file is read in reverse order to complete the job. Other than the direction of scan being different, the two passes function in the same way. .P The global dilation routine for large images is BGDIL and it performs two passes over a globally eroded image to effect a dilation of any number of layers. The end result is to save a significant amount of computation time as compared with the obvious implementations. In fact, any size of opening can be computed in four passes through the image; this includes the overhead of computing the distance map. The iterative method would require 2*S passes to compute an opening of size S. Also, the fast method can compute .i all .r openings of sizes from 1 to 10 in 22 passes, as compared to 218 passes for the obvious iterative method. .P The erosion/dilation procedure described here has also been applied to microscope images from biological sources and as a pre-processing stage on raster map images before extracting vectors (Parker 1989). .ls 1 .sp 1 .ne 8 .b .ce 1 4. References .r .sp 1 .ip

[1] Calabi, L. and Hartnett, W.E., .b Shape Recognition, Prairie Fires, Convex Deficiencies and Skeletons, .i Am. Math. Monthly, .r Volume 75, 1968. Pp. 335-342 .ip [2] Daley, P.F., Racscke, K.,Ball, J.T., Berry, J.A., .b Topography of Photosynthetic Activity of Leaves Obtained From Video Images of Chlorophyll Flourescence, .i Plant Physiology, .r Vol. 90 No. 4, August 1989. Pp 1233-1242. .ip [3] Ehrlich, R., Kennedy, S.K., Crabtree, S.J., and Cannon, R.L., .b Petrographic Image Analysis of Reservoir Pore Complexes, .i Journal of Sedimentary Petrology, .r Vol. 54, No. 4, December 1984. p. 1365-1378. .ip [4] Marshall, S., .b Review of Shape Coding Techniques, .i Image and Vision Computing, .r Vol. 7 No.

4, November 1989. Pp 281-294. .ip [5] Matheron, G., .b Elements Pour Une Theorie Des Milieux Poreux, .i Mason, .r Paris. 1967. .ip [6] Matheron, G., .b Random Sets and Integral Geometry, .i Wiley, .r New York. 1975. .ip [7] Parker, J.R., .b A Faster Method for Erosion and Dilation of Reservoir Pore Complex Images, .i Canadian Journal of Earth Sciences, .r Vol 25, 1988. Pg 1128-1131. .ip [8] Parker, J.R., .b Extracting Vectors from Raster Images, .i Computers & Graphics, .r Vol 12 No 1. 1988 Pp 29-36. .ip [9] Pavlidis, T., .b Algorithms for Graphics and Image Processing, .i Computer Science Press, .r Rockville, MD., 1982 Pp. 199-208. .ip [10] Pratt, W. K., .b Digital Image Processing, .i John Wiley & Sons, .r New York, 1978. Pp. 519-520. .ip [11] Rink, M., .b A Computerized Quantitative Image Analysis Procedure for Investigating Features and an Adaptive Image Process, .i Journal of Microscopy, .r Vol. 107, Pt. 3, August 1976. pp 267-286. .ip [12] Rosenfeld, A., Kak, A.C., .b Digital Picture Processing, .r Vol. 2 (Second Edition), .i Academic Press, .r New York. 1982. .sp 1 .ip [13] Serra, J., .b Image Analysis and Mathematical Morphology, .i Academic Press, .r London. 1982. .ip [14] Silage, D.A., Gil, J., .b Morphometric Measurement of Local Curvature of the Alveolar Ducts in Lung Mechanics, .i Journal of Applied Physiology, .r Vol. 65 No. 4, October 1988. Pp. 1592-1597. .ip [15] Smeulders, A.W.M., ten Kate, T.K., .b Accuracy of Optical Density Measurement of Cells. 1: Low Resolution, .i Applied Optics, .r Vol. 26 No. 6, August 1987. Pp 3249-3257. .ip [16] Sternberg, S.R., .b Esoteric Iterative Algorithms, .r in .b Digital Image Analysis .r (S. Levialdi, Ed.), .i Pitman Publishing, .r 1982. Pp 60-68. .ip [17] Young, I.T., Peverini, R.L., van Otterloo, P.J., .b A New Implementation For The Binary and Minkowski Operators, .i Computer Graphics and Image Processing, .r 17, 1981. Pp 211-224.

[18] Young, I.T., Walker, J.E., Bowie, J.E., .b An Analysis Technique for Biological Shape, *Information and Control*, Vol. 25, 1974. Pp 357-370.