Chapter 3

An Overview of Image Processing and Analysis Techniques for Morphometrics

F. James Rohlf

Department of Ecology and Evolution
State University of New York
Stony Brook, NY 11794

Abstract

This is a general introduction to methods for image processing and image analysis that are useful in morphometrics. Image processing consists of methods to enhance images, such as contrast enhancement, filtering, edge detection, etc. so that the desired details of the images are more evident, especially when viewed by a human. Image analysis is concerned with automatically isolating objects in the image and then obtaining descriptive information about the objects. Alternative sets of features may be mathematically equivalent in their ability to describe an object, but analyses based on different features may give different results. Some implications of this for morphometrics are also discussed.

Introduction

This paper cannot replace a detailed text on image analysis, but it should to serve as an introduction to those image processing and image analysis techniques that are useful (or are expected to become useful) in morphometrics. In addition, it considers some of the implications of the fact that large numbers of new kinds of morphometric characters are available once images of the organisms have been captured and manipulated by computers.

Models for the image-forming process itself are covered first. These are needed in order to understand some of the kinds of information present in an image as well as sources of distortion. The computer hardware involved in the scanning process is not discussed because it is covered elsewhere in this volume (chapters by Fink and by Macleod). The types of techniques available for enhancing an image to minimize the effects of known kinds of distortion are described as well as methods that transform the image to accentuate its desirable aspects. These operations, in which new images are created from old images, correspond to the field of image processing. The field of image analysis is concerned with methods for breaking a scene into its components (at least into object versus background), extracting useful descriptive information about the objects in the image, and interpreting this information (recognition of the objects and their relationships to one another).

The following texts are especially helpful general introductions to image processing and image analysis: Ballard and Brown (1982), Horn (1986), Pavlidis (1982), and Rosenfeld and Kak (1982). Journals that publish technical papers in this field include: Computer Vision; Graphics and Image Processing; and I.E.E.E. Transactions on

Pattern Analysis and Machine Intelligence. The former publishes Rosenfeld's extensive annual reviews of image processing and image analysis literature (the bibliographies usually have over 1,000 entries).

Image Geometry and Image Functions

A basic understanding of how an image is formed is important for an understanding of the methods used to obtain information about the geometrical form of the original object being studied. An image is treated as a two-dimensional pattern of brightness that is produced by an optical system such as a camera. An ideal pin-hole camera is the simplest model of the relationship between points on the object and points in the image (see Figure 1). Since light travels in straight lines, each point in the image corresponds to a particular ray of light projected back toward the scene containing the object. The direction is defined by the position of the point on the image and the location of the pin-hole. As a result of this geometry, the projection onto the image plane yields a perspective projection. The optical axis is the perpendicular vector from the pin-hole to the image plane (the length of this vector is f'). Consider a point P on the object. To compute its location, P', on the image plane, a coordinate system must be established. It is convenient to use the location of the pin-hole as the origin with the z-axis aligned with the optical axis and pointing toward the image (thus points in front of the camera will have negative z-coordinates). Let the x-axis extend to the right and the y-axis upwards. If the coordinates of P are given by the column vector $P = (x,y,z)^t$, then the vector of coordinates of P' can be found as follows:

$$p' = \frac{f}{p^t z} p, \tag{1}$$

where z is the unit vector along the optical axis. The elements of p' are

$$x' = f\frac{x}{z}$$
$$y' = f\frac{y}{z} \tag{2}$$
$$z' = f \quad .$$

In order for the object scene to illuminate the image plane, the pin-hole must have a finite diameter to permit light to enter, but this leads to a blurring of the image. A solution is to use a lens rather than a pin-hole. When in focus, a perfect lens generates an image that obeys the same projection equations, as given above. The relationship between the focal length, f, of a lens and the distances to the object and the focal plane are shown in Figure 2 and in the equation:

$$\frac{1}{f} = \frac{1}{z'} + \frac{1}{-z}, \tag{3}$$

where z' is the distance from the lens to the image plane and $-z$ is the distance from the lens to the object (as above, z-coordinates are negative in front of the lens). If a point is actually at a distance $\bar{z}$, then it will be imaged as a blur circle of diameter

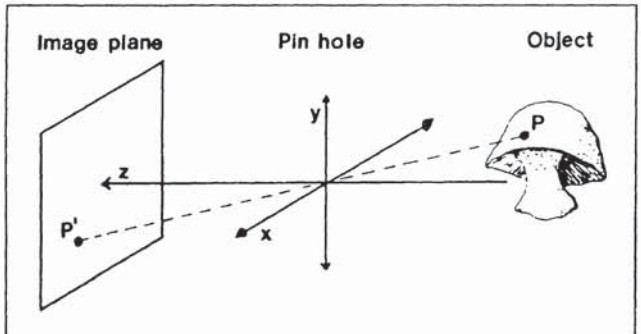


Figure 1. A model for a pin-hole camera. A point P on the object is projected onto a point P' on the image plane.

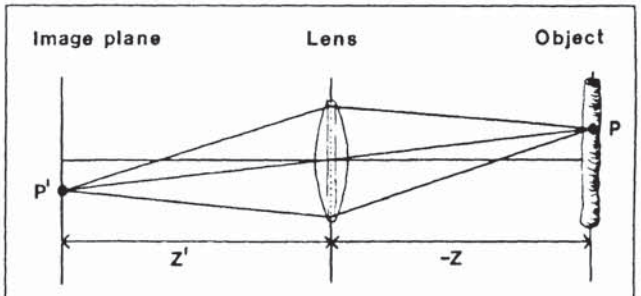


Figure 2. A model for the projection of a point onto the image plane in a camera with a lens. The z-axis is positive toward the image plane.

$$d\frac{|\bar{z}-z'|}{z'}, \tag{4}$$

where d is the diameter of the lens and $\bar{z}$ is the z-coordinate of the point at which $\bar{z}$ is imaged in front or back of the image plane). Thus larger lenses have a smaller tolerance or depth of field.

The brightness, or image irradiance, at each point, (x,y), in the image can be represented by an image brightness function, $f(x,y)$. Irradiance is measured in watts per square meter of radiant energy falling on the image plane. The irradiance of a small area on the image plane, corresponding to a small surface patch at position P on the object, can be computed as

$$E = L\frac{\pi}{4}\left[\frac{d}{f}\right]^2 \cos^4\alpha, \tag{5}$$

where L is the scene radiance of the object surface in the direction of the lens, d is the diameter of the lens (see below), f is the focal length of the lens, and α the angle between the vector p and the optical axis (see Figure 3). Therefore image irradiance is proportional to scene radiance.

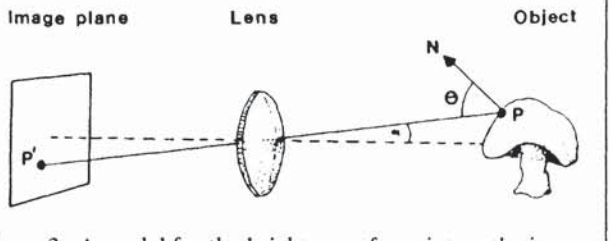


Figure 3. A model for the brightness of a point on the image plane as a function of the orientation of a patch on the surface at the corresponding point on the object.

Scene radiance, L, is power per unit foreshortened-area emitted into a unit of solid angle and is measured as watts per square meter per steradian. It is more complex to model accurately, but there are several important generalities. The foreshortening effect is proportional to $\cos\theta$, where θ is the angle between the surface normal and the vector p toward the lens. Thus, less light is directed toward the lens if the surface is directed *away* from the lens. Unless the surface is matte (an ideal Lambertian surface that appears equally bright from all viewing directions and reflects all incident light), some light will also be reflected towards the lens. This will cause the object to appear glossy or mirror-like (specular). One usually wishes to minimize the effects of scene irradiance so that reflectance, which is a property of the object itself, can be measured. This can best be done by making sure that the objects of interest are evenly illuminated. If this is not possible, one can try applying various mathematical corrections to the resultant image. In the models shown above, image irradiance is a function of the product of a number of factors. Since most of the methods for image enhancement involve only linear operations, it is useful to use log-transformed brightness values as input for the methods described below.

Color

Color will not be considered in this review except to note that digitization of an image at more than one wave length captures more information about a scene. Color images require additional storage space and processing power in a computing system but having multivariate information at each picture point can enable more powerful techniques to be used to discriminate among different objects in a scene.

Blurring

Ideally, a camera's optics map a point in the scene into a point in the film, but in practice the point of light is spread out (blurred) as a result of the lens being slightly out of focus, diffraction rings, film grain, the camera not being perfectly steady, etc. The function that describes how a point of light is spread out is called a point-spread function (it is not imaged as a distinct small circle as implied by the equation given above). The spread of brightness values is often approximated by a normal curve. The effect of a point-spread function at a given point in the final image can be modeled by superimposing point-spread functions at each point in the image (with the height of each point-spread function being proportional to the brightness of the

input point). The resulting brightness in the final image is the sum of the heights of the point-spread functions at each point. For a 1-dimensional image, this corresponds to

$$h(y) = \int_{-\infty}^{\infty} f(y-x)\, g(x)\, dx, \tag{6}$$

where f corresponds to the point-spread function, g corresponds to the input function, and h is the resultant image function. This operation is called the convolution of the functions f and g and is symbolized as $f \otimes g$. The function f is called the kernel of the convolution.

The 2-dimensional generalization is

$$h(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-u, y-v)\, g(u,v)\, du\, dv. \tag{7}$$

It can be shown that the convolution operation is both associative and commutative, $f \otimes (g \otimes h) = (g \otimes f) \otimes h$ and $f \otimes g = g \otimes f$. The operation is well-behaved and easy to work with.

Spatial and Frequency Domains

The input function, $f(x,y)$, can be modelled as the sum of an infinite number of sinusoidal curves. This allows the input function to be expressed as

$$f(x,y) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, e^{i(ux+vy)}\, du\, dv, \tag{8}$$

where

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-i(ux+vy)}\, dx\, dy \tag{9}$$

$F(u,v)$ is called the Fourier transform, $\mathcal{F}$, of $f(x,y)$. While $f(x,y)$ is always real, $F(x,y)$ is generally complex.

Certain operations are more easily performed on the Fourier transformation of a function than on the function itself. For example, it can be shown that the Fourier transform of the convolution of two functions is simply the product of the Fourier transforms of each function considered separately.

$$\mathcal{F}(f \otimes g) = FG, \tag{10}$$

where $\mathcal{F}(f) = F$ and $\mathcal{F}(g) = G$. Not all functions have a Fourier transform. Other difficulties are that the integrals are taken over the entire x,y-plane, whereas imaging devices produce images for only a finite part of the image plane: also digital computers must use discrete samples of these images. For an image with M,N rows and columns, the discrete version is

$$F_{mn} = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} f_{kl}\, e^{-\pi i(km/M + ln/N)}, \tag{11}$$

for $0 \le m \le M-1$ and $0 \le n \le N-1$. Its inverse transform is

$$f_{kl} = \sum_{m=1}^{M-1}\sum_{n=0}^{N-1} F_{mn}\, e^{\pi i(km/M + ln/N)}, \tag{12}$$

for $0 \le k \le M-1$ and $0 \le l \le N-1$. These expressions can also be given in terms of sine and cosines (which is more common in the morphometric literature), rather than as exponentials of complex numbers (which is more compact), using the Euler relation

$$e^{iu} = \cos u + i \sin u. \tag{13}$$

The use of the Fourier transform assumes that the image is doubly periodic (replicates of the image repeat in both the x and y directions). Unless the image at the left edge happens to match that at the right edge (and the top also matches the bottom edge), there will be a discontinuity and some high-frequency components will be introduced. This problem can be avoided by making sure that

there is a uniform background all around the object and that the entire object is within the image.

Digital Images

Of course, the actual images processed by digital computers must be represented as discrete samples of the image brightness surface over a finite range. The image is represented as a 2-dimensional array of measurements of brightness. This array usually has about 500 rows and columns (but devices are available that provide greater resolution). Each element of the array is an integer, usually recorded to 8 bits of accuracy, giving the average brightness of a small region in the image. This element is called a *pel* or *pixel* (short for "picture element"). Thus digital images can be treated as 2-dimensional tables of numbers. Figure 4 shows an image as an image surface; the brightness values for selected rows in the digitized image are plotted as a function of column position.

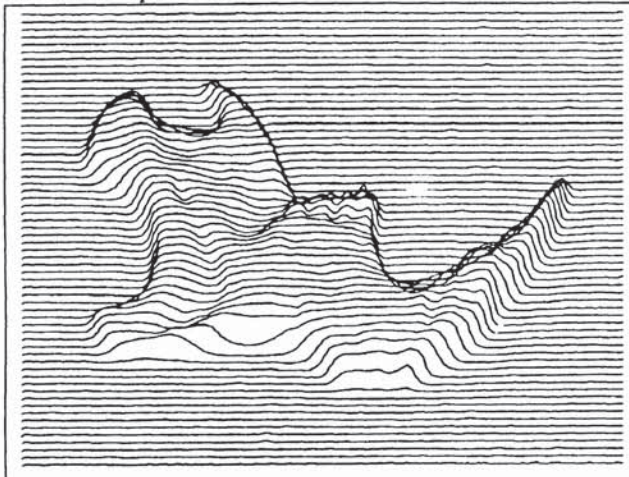


Figure 4. Digital image of a mouse mandible shown as an image surface. Brightness values for the selected rows plotted as a function of column position.

Problems of Sampling

Using a discrete, rather than a continuous, image introduces an effect called aliasing. It can be shown (e.g., Horn, 1986) that sampling an image function, $f(x)$, at intervals of Δx in the image (the spatial domain) is equivalent to replicating the Fourier transform of the image function, $F(x)$, at intervals of $1/\Delta x$. If there are frequencies in the original image greater than $1/\Delta x$, then components of F will interact to produce a composite image transform, F'. Basically, sampling causes information at high spatial frequencies to interfere with that at low frequencies (see Figure 5). This phenomenon is called aliasing, since a wave of frequency $\omega > A$ produces the same wave in the sample as a wave with frequency $2A-\omega$. Therefore, the image should not contain frequencies smaller than half the sampling frequency if this problem is to be avoided (this lower threshold is called the Nyquist frequency). But some objects are better recognized at lower resolutions (where the effects of high frequency noise is averaged out).

One way to reduce the effects of aliasing is to use a pyramidal image data structure (see below), where the search for structure begins at low resolution and then resolution is increased as needed. Rather than redigitizing at lower resolutions (which would introduce aliasing), the lower-resolution images are computed as averages from the original high-resolution image. The consolidation that takes place as one creates lower-resolution images tends to offset the aliasing that would be introduced if one were to digitize at larger sampling intervals. The averaging attenuates the higher frequencies involved in aliasing. Algorithms have been developed to perform many types of image processing operations directly on data stored in a pyramid.

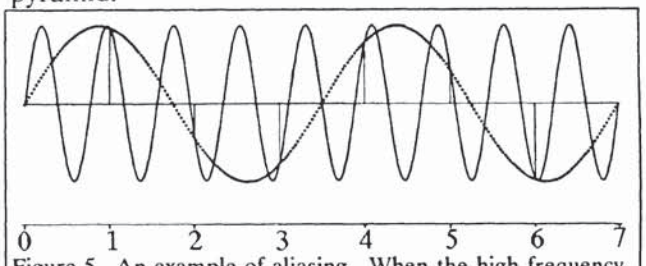


Figure 5. An example of aliasing. When the high frequency wave is sampled at a lower frequency the observed data points appear to have been sampled from a lower frequency wave.

Metrics

As described above, the usual scanning hardware produces a rectangular array of brightness values.

This rectangular spatial pattern is convenient for storage and indexing in digital computers, but it complicates the interpretation of the topological relationships among objects in the image. These considerations are important for the description of outlines using chain codes. The Jordan curve theorem states that a simple closed curve should separate an image into two simply-connected regions.

But consider the following binary image where the "0" state corresponds to the background:

0	1	0
1	0	1
0	1	0

If we adopt the principle of 4-connectedness (a point is considered adjacent only to its immediate neighboring points, left, right, above, and below it) then the four objects, "1", do not form a closed curve, yet the background cell in the center is not connected to the rest of the background. We thus have two background regions without a closed curve. On the other hand, if we adopt 8-connectedness (a point is also considered adjacent to its diagonal neighbors) then the four object cells form a closed curve but there is now only one background region because the center cell is now connected to the other background cells. One solution is to use the 4-connectedness principle for objects but the 8-connectedness rule for background (or vice versa). Horn (1986) suggests a type of 6-connectedness. In addition to up, down, left, and right, he considers cells to be neighborsif they are diagonally above and to the left or below and to the right. Of course, he could have arbitrarily chosen the diagonal directions above and to the right and below and to the left. If one had a hexagonal array, all six cells touching a particular cell would be considered neighbors. This would be much simpler, but standard hardware gives only rectangular arrays.

Data Structures for Digital Images

High-resolution images require considerable amounts of storage (a standard 480×512 8-bit image requires 245,760 bytes). That makes it important to use efficient methods for the storage and retrieval of images. The basic approach is to take advantage of the fact that the brightnesses of adjacent pixels are not independent of one another but are usually similar (the phenomenon of spatial coherence). There are two aspects to efficiency: compactness of storage and speed of retrieval of information. Both aspects are important. The techniques described below emphasize compactness of storage, since that is usually the limiting factor on the microcomputers most often used in morphometrics.

Run length encoding This is a simple, and often very effective, technique for the efficient storage of images that have only a few levels of brightness (e.g., binary images). The image is stored as a continuous stream of bytes but with the start of each row marked with a special code (or else the byte offset of the start of each row stored in a separate array). The lengths of each of the runs of identical brightness values along each row are stored rather than the actual brightness values. For example, a row of brightness values might be:

0	1	1	1	1	0	0	0	0	0	0	1	0	1	1	0

The runs would then be (1, 4, 6, 1, 1, 2, 1). Note: in order to recover the brightness values each row must begin with the same brightness value, say 0. If a line happens to begin with a brightness value of 1 then an initial run of length 0 is inserted at the beginning of the row. Some image operations can be performed on the image in this compressed format. For example, the area of the object (the 1's in the image) is simply the sum of the even-numbered runs. Ballard and Brown (1982, pp. 58-61) show how to compute the horizontal and vertical projections of an image and its center of gravity from this storage representation.

Pyramids In some applications it is useful to be able to process an image at varying degrees of resolution. One method is to partition the digitized image into non-overlapping regions of equal size and shape and then to replace each of these regions by the average pixel density in that region. This step is called consolidation. This is repeated recur-

sively until there is only a single region with a brightness value equal to the average brightness in the original image. Using the average brightness value for a region rather than a value from the center of the region tends to reduce the aliasing effects that one would expect if one were just to redigitize the image at a lower resolution. Ballard and Brown (1982, pp. 109-111) show an example (from Tanimoto and Pavlidis, 1975) of an algorithm for edge detection using data stored in a pyramid.

Quad-trees Quad-trees (Samet, 1980) are an efficient method to store binary images. To convert data stored in a pyramidal data structure to a quad-tree one recursively searches the pyramid from top to bottom. If an element is "black" or "white" then form a terminal node in the quad-tree of the corresponding type. Otherwise, form an internal "gray" node with pointers to the results of the recursive examination of the 4 elements at the next level in the tree. Samet (1981a) gives an algorithm to directly convert a raster image (the usual storage is by rows) to a quad-tree. The method is efficient in that the image is read and processed a row at a time and the resulting quad-tree is of minimal size. Less space is needed by the algorithm than would be required if the entire image were read at once. There are many algorithms to perform image processing operations directly on images stored as quad-trees. The computation of the area of an object is easy. Samet (1981b) gives an algorithm for computing the perimeter of regions from a quad-tree representation. Samet (1983) shows how to perform a medial axis transformation from a quad-tree. A number of papers have been published describing the computation of various geometrical properties of objects from a quad-tree representation. Pavlidis (1982) is a convenient source of many of these algorithms.

One problem with quad-trees is that they are not translation independent—if an object is shifted in position by one pixel the quad-tree can be very different in structure. Scott and Iyengar (1986) developed a translation invariant version of the quad-tree based upon the medial axis transformation (see below).

Filters

A filter is a function which produces new images that are transformations of an input image. The purpose is to produce images in which particular aspects of an image are accentuated or enhanced. The results can sometimes be quite impressive (such as revealing details hidden in shadow areas of a photograph, or the apparent sharpening of an out-of-focus image). Of course, the desired information must already be present in the image. What the enhancements do is transform the information so that the desired features of the image are more obvious to a human observer. It is useful to imagine an image as a surface where height represents the brightness at each point in the image. An example is shown in Figure 4 above, where each horizontal curve corresponds to a column in the digitized image. A transformation of such an image surface might, for example, smooth out the part of the surface corresponding to the background but steepen the sides of the hills corresponding to the object, so that it appears to have sharp vertical cliffs. The published literature on methods of enhancement is extensive and growing rapidly. Some standard methods are described below. These transformations cannot perform "magic." A great deal of effort can be saved by starting with simple images of well-illuminated scenes.

Contrast Enhancement

One of the first adjustments to consider is contrast enhancement. Brightness values are rescaled so that they cover the full dynamic range of the display device. For example, in a very low contrast image the brightness values may range from 150 to 200. Rescaling them to range from 0 to 255 will result in an image that humans find easier to interpret. If the digitizing hardware can perform this operation as it is acquiring the image, the researcher is able to make more efficient use of the resolution of the digitization of the brightness values. This is especially important if the digitizer cannot furnish at

least 256 brightness levels. Contrast enhancement is useful even when working with an already digitized image. However, spreading out a range of 150-200 to 0-255 does not increase the number of distinct brightness values; they are simply spaced further apart. The operation is still useful, however; the image is more pleasant to look at even though it contains no more information than the original.

Histogram Transformations

Histogram enhancement techniques include contrast enhancement, as described above, but also make a non-linear transformation of the brightness values so that the brightness values not only cover the full dynamic range but have a frequency distribution of brightness values that take on a particular form. In theory, a uniform distribution can be obtained using the following transformation

$$g(q) = \frac{M}{N} \int_0^q h(p)\, dp, \qquad (14)$$

where M is the number of gray levels, N the number of pixels, and $h(p)$ the observed histogram of the number of pixels with each level of brightness p. The practical problem with this algorithm is that the input brightness values are discrete, so that the most we can do is to obtain a more even spacing of the values. The histogram need not be uniform since some classes may have many more entries than others. The results can be rather disappointing when the important information in the image is best represented by only a few distinct brightness values.

Hummel (1977) suggested a transformation called histogram flattening in which the histogram is made more uniform by randomly assigning pixels in the most abundant classes to other brightness classes. He suggested that the choice of pixels to be reassigned could be based upon the average gray-level in their local neighborhoods. On the other hand, Frei (1977) suggests that, for images to be interpreted by humans, the goal should be to produce a picture in which there is a uniform distribution of perceived brightness levels. To do this, the distribution of displayed brightness levels should be hyperbolic rather than uniform. But this seems to apply only to images of objects that the viewer expects to see represented by continuous brightness levels. Simple high-contrast images (such as bone or shells laid on a sheet of black paper) are not improved by such transformations.

Adaptive contrast enhancement methods adjust the degree of contrast enhancement of a pixel depending upon the distribution of brightness of its neighboring pixels. Peli and Lim (1982) proposed a method in which the final image is a weighted combination of a smoothed image and the difference between the original image and the smoothed image. The weights can be a non-linear function of the brightness in the smoothed image so as to give greater increase in contrast to certain ranges of brightness values. An examination of the gray-level histogram can also be useful when trying to find a suitable threshold level to segment the image into regions. If the scene consists of just an object and background, then one would hope to find two peaks and the threshold would be placed in the valley between them. The results are not as clear-cut as one might expect. One problem is that pixels on the boundary of an object are expected to have intermediate gray values dependent upon their degree of overlap with the object versus the background. Other problems are shadows, uneven illumination, and noise.

Smoothing

Smoothing is a useful technique to eliminate unwanted fine detail in an image by averaging each pixel's brightness value with those of its neighbors. Such methods are sometimes referred to as low pass filters since they remove high-frequency details (fine undulations in the image surface) while preserving the low frequency information (large scale changes in the surface). Often the general form of an object is of interest rather than its fine details (texture). A simple analog method of smoothing is to digitize the image when it is slightly out-of-focus. Mathematically, smoothing corre-

sponds to the convolution of the image function with a point-spread function; the brightness at each pixel is spread-out and averaged with adjacent pixels. This results in a linear, space-invariant (the same function is used over the entire image), moving average (each pixel is some type of average of its neighbors) filter. This can be implemented using the following equation:

$$g(x,y) = h \otimes f$$

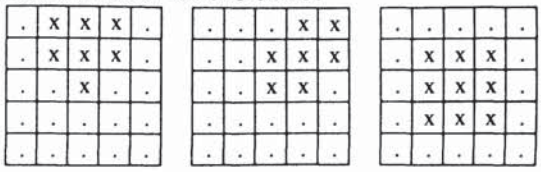
$$= \sum_{u} \sum_{v} h(u-x, v-y)\, f(u,v)', \qquad (15)$$

where h is the point-spread function (the kernel of the convolution), f is the input image function, and the summations are overall alignments at which the kernel overlaps the given pixel, at x,y.

There are many choices for h depending upon the type and degree of smoothing desired. A common choice is the values in the following array.

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

Usually this filter does not remove much noise from an image. To produce a stronger effect, one could either use a larger array of constants or else apply the filter repeatedly. Such simple local smoothing applied to the entire image often removes some important details in the image since the brightness values along the outline of an object will also be averaged with those of the background. This makes the boundaries of an object more difficult to detect by the human eye. A solution is to limit the smoothing to regions of relative homogeneous brightness levels. Nagao and Matsuyama (1979) proposed that one examine subregions around each pixel and then average its brightness value only with those in the most homogeneous subregion (an edge-preserving smoothing transformation). The procedure computes the variance in regions corresponding to the x's in the 4 possible 90° rotations of each of the first two patterns (within a 5×5 region centered on each pixel) and the one possible orientation of the third.



Nagao and Matsuyama (1979) suggested using the ordinary variance as a criterion of homogeneity. I have found slightly better results by weighting the center point when computing the mean and variance. One could also include a tolerance so that no averaging would take place if even the most homogeneous region was too heterogeneous. The method is time-consuming since nine variances must be computed at each pixel. If one knows that the gray-values in a particular region of an image should be uniform, one can limit the smoothing operation to that particular area and thus avoid the smoothing of edges.

Background Subtraction

In some applications it is possible to get rid of some of the complexity of the image (dirt spots on the lens, etc.) by subtracting out a constant background. If the images are correctly aligned, one can simply subtract the gray-values of the background image from that of the image under study.

Template Matching and Cross Correlation

A very common operation is that of matching a template pattern against an image. On a 1-dimensional image one might slide the template pattern, t = (-1, 0, 1), across an image, looking for the position of greatest match (which, in this example, is the position of the largest linear increase in image intensity). The distance between the image and a template (aligned at pixel y) is

$$d_y = \sqrt{\sum_{x} (f(x)-t(x-y))^2}, \qquad (16)$$

where the summation is over all pixels in the image for which the template is defined in this alignment. Finding the location that minimizes d_y is equivalent to finding the location that maximizes the cross correlation function, R_{ft}, for f and t.

$$R_{ft} = t^*f = \sum_x f(x)\, t(x-y) \quad . \tag{17}$$

Note the similarity of this function to the convolution operation [in the convolution $t(x-y)$ is replaced by $t(y-x)$]. The same operation can be applied to a 2-dimensional image. The 2-dimensional template is "rubbed" over the entire image and a value is computed for each alignment tested. The continuous form of the 2-dimensional cross-correlation function is

$$t^*f = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} t(u-x,v-y)\, f(u,v)\, du\, dv. \tag{18}$$

Edge Detection

The regions of rapid change in brightness values in an image, which often correspond to the boundaries or edges of objects, seem to convey much of the information about the shape and locations of objects in the image. Thus methods that enhance this aspect of an image are of particular interest.

Gradients The most obvious operation to consider is the computation of the magnitude of the gradient of the image surface at each pixel. If the result is viewed as an image, pixels located in regions of rapid change in brightness in the original image would appear as bright points. An object such as a dark leaf against a light background would thus appear as a set of bright pixels around the perimeter of the leaf. Such a transformed image may be simpler to process by computer program so as to obtain particular features of interest (e.g., the number of pixels brighter than a certain threshold could be used as an estimate of the perimeter of the leaf). However a simple gradient computation may fail since it is very sensitive to noise in the image. Thus one may wish to use the gradient of a smoothed image or to use more complex algorithms such as those of Machuca and Gilbert (1981).

Horn (1986) shows a simple model for an edge in an image as a straight line separating two regions of different brightness.

$$E(x,y) = B_1 + (B_2-B_1)\, u(x \sin\theta - y\cos\theta + \rho)\,, \tag{19}$$

where B_1 and B_2 are the brightness values in the two regions, $x \sin\theta - y\cos\theta = \rho$ is the equation of the separating line, and $u(z)$ is the unit step function:

$$u(z) = \begin{cases} 1 & \text{for } z > 0 \\ 1/2 & \text{for } z = 0 \\ 0 & \text{for } z < 0\,. \end{cases} \tag{20}$$

The gradient of this surface is the vector

$$\begin{bmatrix} \partial E/\partial x \\ \partial E/\partial y \end{bmatrix}, \tag{21}$$

where

$$\begin{aligned} \partial E/\partial x &= \sin\theta\,(B_2-B_1)\,\delta(x\sin\theta - y\cos\theta + \rho) \\ \partial E/\partial y &= -\cos\theta\,(B_2-B_1)\,\delta(x\sin\theta - y\cos\theta + \rho). \end{aligned} \tag{22}$$

It is important to note that the gradient is coordinate system independent in that it maintains its magnitude and orientation relative to the underlying edge when the separating line is rotated or translated.

Laplacian The Laplacian of the surface defined above is

$$\begin{aligned} \nabla^2 E &= \partial^2 E/\partial x^2 + \partial^2 E/\partial y^2 \\ &= (B_2-B_1)\,\delta'(x\sin\theta - y\cos\theta + \rho)\,, \end{aligned} \tag{23}$$

where δ' is the unit doublet, the derivative of the unit impulse $\delta(u)$. The Laplacian has the desirable properties of retaining the sign of the brightness difference across the edge, so we can determine

which side is brighter and thus reconstruct the original edge, and it is a linear function of x and y.

Approximations for Digital Images

The simplest approximation is to estimate the derivatives of the surface at each point in the image by the differences in E_{ij} at adjacent pixels. Let the pixels around a point be represented by the following table.

E_{ij+1}	E_{i+1j+1}
E_{ij}	E_{i+1j}

Note that the order of subscripts corresponds to the x and then the y dimension, not the usual row and then column convention with matrix algebra. The derivatives at the center of this 2×2 array can be estimated as

$$\begin{aligned}\partial E/\partial x &= \frac{1}{2\varepsilon}\left((E_{i+1j+1}-E_{ij+1}) + (E_{i+1j}-E_{ij})\right)\\ \partial E/\partial y &= \frac{1}{2\varepsilon}\left((E_{i+1j+1}-E_{i+1j}) + (E_{ij+1}-E_{ij})\right)\end{aligned}, \quad (24)$$

where ε is the spacing between the rows and columns. This formula (from Mero and Vassy, 1975) is the average of two finite-difference approximations and is an unbiased estimate of the slope for the point where the four pixels used in the above formula meet. The squared gradient can be used to produce a map of the location of high rates of change in brightness in the image. To get the actual direction of change one must refer back to the gradient itself. A more refined estimate of the slope of the surface can be obtained using the information from a 3×3 arrays of pixels.

E_{i-1j+1}	E_{ij+1}	E_{i+1j+1}
E_{i-1j}	E_{ij}	E_{i+1j}
E_{i-1j-1}	E_{ij-1}	E_{i+1j-1}

The Sobel estimate of the gradient (Shaw, 1979) is computed by cross correlating the above submatrix of E_{ij}'s with the weights in the following two tables (the first yields the change in the x-direction and the second gives the change in the y-direction).

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

From this 3 by 3 array of pixels we can also estimate the second partial derivatives as

$$\begin{aligned}\partial^2 E/\partial c^2 &= \frac{1}{\varepsilon^2}(E_{i-1j}-2E_{ij} + E_{i+1j})\\ \partial^2 E/\partial c^2 &= \frac{1}{\varepsilon^2}(E_{ij-1}-2E_{ij} + E_{ij+1})\end{aligned}, \quad (25)$$

so that the Laplacian can be estimated as

$$\nabla^2 E = \frac{4}{\varepsilon^2}\left[\frac{1}{4}(E_{i-1j} + E_{ij-1} + E_{i+1j} + E_{ij+1})-E_{ij}\right], \quad (26)$$

This function is zero both in areas of constant brightness and also in areas where brightness varies linearly. It represents subtracting the value of a central pixel from the average of its neighbors. This corresponds to the application of a template with weights $1/\varepsilon^2$ times the values in the following table:

0	1	0
1	-4	1
0	1	0

Note that one could just as logically rotate the coordinate system by 45° before approximating the derivatives. Linear combinations of this and the rotated template also produce estimates of the Laplacian. Horn (1986) states that the following template, times $1/6\varepsilon^2$, is popular and produces a particularly accurate estimate of the Laplacian.

1	4	1
4	-20	4
1	4	1

Kirsch Operators

The Kirsch operator (Kirsch, 1971) is a method for classifying the properties of small regions in an image. It can be used to detect whether a region gives evidence for an edge, a line, or is undifferentiated. For example, to determine the direction of a gradient at each pixel in the image one could compute

$$S(x) = \max [1, \max_{k} \sum_{k-1}^{k+1} f(x_k)], \quad (27)$$

where the $f(x_k)$ are the 8 neighboring pixels to x and where subscripts are computed modulo 8. The value of k that yields the maximum indicates the direction of the gradient (with 3 bits of accuracy). This method can easily be implemented by matching (cross correlating) the following four templates with a span of $n = 1$ (similar templates can be used for larger values of n). The direction is then indicated by whichever template matches best.

-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	-1

0	1	1
-1	0	1
-1	-1	0

1	1	0
1	0	-1
0	-1	-1

Enhancement of Geometric Patterns

Methods have also been developed that accentuate particular geometric patterns in an image, rather than performing more general enhancements. Special attention has been given to line enhancement methods. This is both because linear features are often directly of interest (e.g., veins in insect wings or leaves) and also because linear features are useful as boundaries of objects. This does not mean that an object needs to have flat sides—it need only be relatively smooth. Small regions around the outline of a bone or a shell, for example, can be well approximated by a series of linear edges. Paton (1979) proposed several useful methods for finding linear features. In these methods templates are superimposed at various orientations over each pixel in the image. The templates are such that the presence of a line in the region of a pixel will result in a high cross-correlation. The maximal value obtained for templates centered on a given pixel is used as the output value for the given pixel.

Groch (1982) proposed a procedure for recognizing line-shaped objects in images by trying to follow a large number of lines starting from seed points found by searching along transects through the image. A regional operator is then applied to try to fill in gaps along more or less collinear line segments.

Three-dimensional Images

A detailed discussion of the recovery of 3-dimensional information from images is beyond the scope of the present review as it involves more complex techniques. Information on the 3-dimensional orientation of surfaces can be obtained from their pattern of reflectance (and thus requires information about their surface properties). For certain types of objects, one can gain 3-dimensional information by projecting stripes of light, perhaps from a laser, across the object at known angles. The apparent deflection from a straight line in the image can be related to the shape of the object.

Image Segmentation

It is usually desirable to break an image up into regions, or segments, corresponding to the logical subunits of the original scene. In morphometrics, one wishes to separate the image of an object from its background and perhaps to isolate different components of the image. The regions can then be analyzed separately. In some images the objects to be located are lines. One can adapt the line enhancement operators described above. Groch (1982), for example, used this approach to detect roads in aerial photographs. Biological images often contain linear features. Once an object has been separated from its background, contour-tracing algorithms can be used to trace its outline (see below).

Thresholding to Define Regions

While a human can usually easily recognize the component parts of a complex scene (such as background, outline of a wing, veins of a wing, cells between the veins, etc.) this is often a difficult task to perform automatically (Riseman and Arbib, 1977). With perfect, noise free, images one can isolate an object from a uniform background simply by finding a threshold level of brightness such that all pixels in the background are above or below the

selected value. Castleman (1979, p. 311) gives a simple algorithm that can trace out the boundary between the object and the background for such images. He points out, however, that even small amounts of noise can send the tracking algorithm temporarily or hopelessly off the boundary. This is a problem with mosquito wings, for example, since the margin of the wing and the veins are covered with scales that can become dislodged and appear in unexpected locations when the wing is mounted on a slide. An additional problem is the fact that an image may not be illuminated evenly so that the background may differ in brightness in different parts of the image even after some of the standard image enhancement techniques have been applied. Thus the boundary tracking problem can become rather complex and usually must take into account a priori information about the geometrical properties of the particular class of objects being extracted, or else be supervised by someone who knows what the expected contour should be and thus can intervene and make corrections when necessary.

Region Growing

A complementary approach is that of "region growing" (Brice and Fennema, 1970) in which one first examines small regions in an image and then merges adjacent regions with similar properties (brightness, texture). This has been used, for example, to break aerial photographs into homogeneous blocks each representing a different type of forest, farmland, etc. Grainger (1981) reported an average accuracy of about 50% when this method was applied to 186 sample sites from New Forest, Southern England, for which both ground and densitometric data were available. This method is most often used with multispectral images (grey-scale images at each of several spectral bands). There is much more information for each pixel, and hence higher performance can be expected.

Contour Tracing

The most important technique for image segmentation in morphometrics is that of tracing the outline of a selected object. If the grey-levels of the object are distinct from the background, this is a relatively straight-forward task. Such images can be converted into binary images, where "1" corresponds to the selected object and "0" to the background pixels by thresholding. Pavlidis (1982) gives algorithms for finding the overall contour and also for finding the contours of any holes that may be contained within an object in such a binary image. In order to describe the algorithms, we need to adopt the following standard numbering system to refer to the 8 pixels adjacent to a given point. For example, the point above the given pixel, p_{ij}, is called the 2-neighbor.

3	2	1
4	p_{ij}	0
5	6	7

A contour is defined as the set of all pixels within the selected object that have at least one neighbor that is not part of the object. The strategy is to start with a point in the object whose 4-neighbor is not in the object, and then to trace the outline in a counter-clockwise direction.

1. Choose a point, A, in the contour such that its 4-neighbor is not in the object.
2. Set C = A, S = 6, and set the flag *first* = *true*.
3. While C ≠ A or *first* = *true*, do steps 4 to 10.
4. Set the flag *found* = *false*.
5. While *found* = *false*, do steps 6 to 9 at most 3 times (the purpose of this limit is to avoid looping on objects that consist of only a single pixel).
6. If B, the (S-1)-neighbor of C, is part of the object, then set C = B and *found* = *true*.
7. Else if B, the S-neighbor of C, is part of the object, then set C = B and *found* = *true*.
8. Else if B, the (S+1)-neighbor of C, is part of the object then set C = B and *found* = *true*.
9. Else set S = S + 2, modulo 8.
10. Set *first* = *false*.
11. End.

The algorithm must also be applied once for each hole in the object. When completed, one needs a description of the path traversed. One possibility is simply to list the coordinates of the points C. Another, more compact representation is to store the coordinates of only the first point and then store the neighbor code to indicate the direction taken when moving from one pixel to the next. When the contour is very smooth, further economy can be achieved by storing the derivative of the chain code. The change in direction will usually require fewer bits than the chain code (unless the contour often doubles back on itself). When the chain code sequence or its derivative contains sequences of identical codes, run-length coding can be used to reduce the amount of space needed to store a contour. In run-length coding one replaces a sequence of identical values with a special code and the length of the sequence.

Thinning

Thinning algorithms simplify the representation of the outline of an object by computing an internal skeleton that will contain useful information about the original outline. Straney (this volume) describes several different methods of defining what one means by a skeleton and different algorithms for their computation. They are usually applied to binary images (images that have already been thresholded). One method is to reduce the width of elongated objects in the image by "eating away" at the sides of objects while trying to avoid the deletion of pixels at the ends of the objects. There are several methods for carrying out such operations. Pavlidis (1982) gives the classical thinning algorithm. He also presents a simple approximate thinning algorithm that may be satisfactory for some purposes. Another approach is to compute the medial axis transformation, MAT. In a MAT skeleton, the pixels are located at centers of circles that touch the original outline at more than one place. For example, if the original object is a circle, then the MAT skeleton will be a single point at the center. If one codes each pixel in the MAT skeleton by the diameter of the circle that it represents, the MAT skeleton has the important property that it can be used to reproduce the original outline shape (i.e., the medial axis transform has an inverse). Blum (1973) describes the geometrical properties of MAT skeletons and some implications for shape description in biology. Bookstein et al. (1985) discuss and give examples of its potential application to morphometrics including that of Bookstein (1981). Thin figures seem to be represented well by MAT skeletons—the skeletons seems to have intuitively reasonable shapes. Bookstein et al. (1985) observe that, while it uses only information on the outline, the skeleton often has a structure that seems biologically appropriate. However, the skeleton seems less useful for wider objects. An important problem is that it is very sensitive to noise. Small changes in the outline (e.g., small bumps or indentations) can cause drastic changes in the form of the skeleton. The outline has to be quite smooth in order for a simple skeleton to be obtained.

There also has been some work generalizing the MAT to gray-level images. Dyer and Rosenfeld (1979) describe a simple algorithm to thin gray-scale images. For dark objects, their method changes each dark pixel to the minimum of its neighbors' levels provided this does not disconnect any pair of points in its neighborhood. This process can be repeated until the objects are sufficiently "thin". Wang et al. (1981) define a gray-scale generalization of the medial axis transformation (which they call a MMMAT, for min-max MAT). It allows one to reconstruct good approximations to the original image.

Texture

A general discussion of this topic is beyond the scope of this review as the complexity and diversity of types of surface textures possible in biological images is very large. However, the use of fractal curves and surfaces has attracted increased interest in many fields in the last few years. One type of application that seems useful in morphometrics is the use of fractal dimension as a description of the texture or complexity of an outline of an object.

For example, one can digitize the outline of a leaf at high resolution and then see how the apparent length of the outline changes as a function of the step-size (scale) used to measure the length of the outline. The fractal dimension of the outline curve can then be estimated using the relationship

$$D = \ln \frac{N}{\ln (1/S)}, \quad (28)$$

where S is the step size and N is the number of steps. For a line in Euclidean geometry, a division of a line into segments of length 1/S results in S segments and hence a dimension of $D = 1$. When the outline is highly reticulate its length will be very long when one measures it with a small step size and as a result its fractal dimension will be greater than 1. Vlcek and Cheung (1986) describe the computation of the fractal dimension for several types of leaves. They show that fractal dimension is a useful descriptor of the irregularity of the leaf outline. The obtained values range from 1.02 for a rather smooth American basswood leaf to 1.28 for a white oak leaf. Long (1985) used fractal dimensionality to describe complex sutures in deer skulls and in ammonites. He found D-values from about 1.4 to 1.5. Morse et al. (1985) found D-values of about 1.5 for the outlines of a variety of plants during early spring. They point out that if insects and other arthropods living on these plants perceive the amount of space on the plant (for food and shelter) in relation to their body size, then small insects will perceive a much larger available habitat than larger insects when $D > 1$. They then show that the distribution of sizes of insects is in keeping with what one would expect if their abundance were proportional to the perceived amount of available habitat. Katz and George (1985) furnish a program in BASIC that estimates the fractal dimension of an outline represented by a set of x,y-coordinates. Slice and Gurevitch (in preparation) used this approach and found significant differences between species and trees of the genus Acer (Maples) with respect to leaf outline complexity. They found that the ordering of species with respect to mean fractal dimension was consistent with their subjective perception of outline complexity. D. Slice has developed a program, called FRACTAL-D, that performs these computations.

A similar idea holds for surfaces. In an Euclidean plane the subdivision into cells with a mesh size of $1/S$ will result in a surface area composed of $N = S^2$ equal sized cells, and hence a dimension of $D = 2$. As the surface becomes more complex, the surface area will increase and hence the fractal dimension will be larger than 2.

Boundary Representation

This is a very important topic. Most applications of image analysis to morphometrics have been concerned with the comparison and analysis of information that can be extracted from an outline (often supplemented by information on locations of morphological landmarks). But in order to use an outline in a quantitative analysis it is necessary to use an appropriate mathematical representation. Listed below are some of the most common approaches (some have been mentioned above). These approaches will not be described in detail since most of them will be covered elsewhere in this volume.

1. x,y-coordinates. One can simply save enough of the coordinates of enough points around the outline to capture its form with sufficient accuracy. Usually one will have more points in regions of higher curvature. Some methods of morphometric analysis use these coordinates directly. These raw coordinates can be used to derive other representations of the outline. For example, the elliptic Fourier method uses coordinates as input (rather than polar coordinates or tangent angles as in most Fourier studies).
2. Chain codes. The method of using chain codes (and differential chain codes) was discussed above.
3. Polar coordinates. If the outline is a simple convex shape, then it may be possible to describe the shape by giving the radius of

equally spaced vectors from some convenient origin to points along the outline contour. This method has been used in many Fourier applications.

4. Tangent angle. While traversing the outline of an object, one can record the slope of a tangent to the outline at the current position and the distance traveled along the outline. This has the advantage that one can represent tangent angle as a function of arc length for any closed outline shape. This has also been used in many Fourier studies.
5. Medial axis skeleton. Since the original outline can be recovered from a skeleton, the skeleton can be used as a method to encode an outline shape.
6. Splines. Several studies have explored the usefulness of using splines rather than Fourier functions to describe the shapes of morphological structures. Some examples are Engles (1986) and Evans et al. (1985).
7. Fractals. Barnsley et al. (1986) show that it is possible to determine the fractal curve that provides a close approximation to a given binary image. Remarkably, this representation required very few parameters to be estimated in order to fit the outline of complex objects with very complex outlines such as a black spleenwort fern frond.

Feature Extraction

Feature extraction is the task of obtaining the most important descriptive parameters from an image. These parameters represent not just an encoding of an image, but the isolation of particular parameters that can be used to distinguish an object in one image from another. These may consist of the usual distance measurements used in morphometrics (lengths, maximum widths, etc.), often the problem is more complex. Traditional measurements are often selected because they are easy to make using hand-held calipers. But with automation other types of measurements may be easier to program in a computer. For example, once one has an outline contour it is easier to measure the area or the perimeter of an object than it is to measure its width. Thus with the availability of new technology one should not just duplicate conventional methods but explore other ways of describing differences among organisms. There are a large number of ways in which an object in an image can be described. Unless the different methods are linearly related, one does not expect them to give exactly the same results. Thus the choice of types of descriptors used in a morphometric analysis is expected to make a difference (see further discussion below). Unfortunately, it is unclear at this point how one should choose among the different systems. But in the important special case of systems of linear distance measurements, Strauss and Bookstein (1982) point out the advantages of taking measurements in the form of a "truss" rather than in the more conventional pattern that often has a lot of redundancy. One of the most popular approaches in morphometrics has been the use of Fourier coefficients to describe outlines of organisms. My comments on this topic is very short since it is covered elsewhere in this volume. The method of moment invariants has been used in a few morphometric studies. It is described in some detail below since different formulations of the method have been used and they raise some interesting issues.

Description of an Outline Contour

A common approach is the fitting of some mathematical function to the points sampled around the outline of an object. The parameters of the fitted function are then used in multivariate analyses as descriptors of the shape of the outlines. Various types of Fourier analysis are the most popular examples of this approach in morphometrics, but other functions have also been used. A brief outline is furnished below with references to more detailed accounts.

1. *Fourier analysis of an outline expressed in polar coordinates.* In many morphometric studies points are sampled along the outline such that

vectors connecting them to some point of reference (or origin) are separated by equal angles. The lengths of these vectors (distances of each point to the origin) are then subjected to a Fourier decomposition (a 1-dimensional Fourier transformation). The resulting coefficients can be expressed in one of two ways: either as the coefficients of the sin and cosine terms in the Fourier series or in terms of their amplitude and phase angle. Kaesler and Waters (1972) provides an early example. When no landmarks are available it may not be possible to specify a unique starting point for the measurement of the angles (i.e., the vector that corresponds to an angle of 0). In such cases, only the amplitudes for each harmonic are used as descriptors. Younker and Ehrlich (1977) provide an example. A limitation of the use of this polar representation is that the outline of the object must be such that each vector crosses the outline only once. Thus the outline cannot be very complex.

2. *Fourier analysis of an outline expressed in terms of the change in tangent angle as a function of arc length.* Bookstein et al. (1982) refer to this as an intrinsic representation. Zahn and Roskies (1972) suggested that an outline be scaled so that its length is equal to 2 and then the following function computed for each point along the outline

$$\phi^*(t) = \theta(t)-\theta(0)-t, \qquad (29)$$

where t is the distance along the outline of a given point, $\theta(t)$ is the angle of a tangent line at that point, and $\theta(0)$ is the angle of a tangent line at the starting point of the outline. Thus $\phi^*(t)$ is the difference between the cumulative change in angle that one observes when moving along an outline and the change that one would expect if the outline were a perfect circle. The values are then subjected to a Fourier decomposition. This approach has the advantage that it can be used for any closed contour (complete outline), regardless of its shape.

3. *Elliptic Fourier analysis.* Kuhl and Giardina (1982) proposed the separate Fourier decomposition of the differences in the x and y-coordinates as a function of arc length corresponding to the distance along the outline to each point (again scaled so the perimeter of the outline is 2π). Rohlf and Archie (1984) showed that this method has several advantages over the methods listed above.

4. *Splines and other functions.* Any function that can be made to pass through an observed set of points can be used as a description of an outline. Cubic splines and Bezier curves represent flexible families of curves that can be made to fit arbitrary configurations of points. Examples of the use of Bezier curves are given by Engles (1986) and examples of cubic splines are given by Evans, et al. (1985).

5. *Eigenshape analysis.* Lohmann (1983) showed that an outline, represented in his case by the $\phi^*(t)$ function, can be fitted by sets of empirical functions derived from the data. The advantages of this approach are that fewer functions are needed to describe the observed diversity among the objects under study, and it is not necessary to specify particular families of curves to be fit to the outlines (such as sums of sines and cosines or various types of polynomial functions). The Chapter 6 by Lohmann and Schweitzer in this volume is a general exposition of this method with examples.

6. *Fractals.* As mentioned above, Barnsley et al. (1986) have shown that it is possible to solve for the fractal curve that best approximates a given object outline. In their examples, very few parameters were needed to obtain a very close fit for objects with very complex outlines. In the case of a black spleenwort fern frond, the outline was described by a collage of four affine transformations which required 28 parameters, 8 of which were zero. So few

parameters were probably required because the frond does seem to be a case where the outline shows the property of self-similarity. Objects with less regularity may require many more parameters. The relevance of these functions for morphometrics needs further study. An important question is the extent to which objects with similar values of the parameters (i.e., those that are close together in the feature space) are similar morphologically.

In all of these cases, the results are a set of coefficients that can be used as measurements of descriptive variables for various types of multivariate analyses. Unfortunately, the results one obtains from multivariate analyses need not be the same for different types of shape descriptors. The descriptors obtained from different methods do not represent simple linear transformations of the same information. The relationships between some pairs of methods correspond to complex non-linear transformations of the original coordinate data. This implies that it is not sufficient for a method to be convenient computationally. In order to use a method one must be confident that it is appropriate for the description of the kinds of variation that ones expects to observe.

Moments of an Image Surface

One simple approach to the description of an image is to transform the image so that the brightness of the background is zero and the brightness values for the object are positive numbers. Then the brightness values can be treated as proportional to a 2-dimensional frequency distribution (a sample from a 2-dimensional probability density function). The 2-dimensional moments of this function can then be computed and used as the parameters of this distribution. For example, the mean in the x-direction is

$$\bar{x} = \int_{-\infty}^{\infty} x\, f(x,y)\, dx\, dy. \qquad (30)$$

The p, q central moment can be computed as

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p\, (y-\bar{y})^q\, f(x,y)\, dx\, dy. \qquad (31)$$

The order of a moment is the sum $p+q$.

A uniqueness theorem (Papoulis, 1965) guarantees that if $f(x,y)$ is piecewise continuous and has nonzero values in only a finite part of the x,y-plane (true by definition for brightness surfaces), then moments of all orders exist, the moment sequence is uniquely determined by $f(x,y)$, and the moments uniquely determine $f(x,y)$. Thus the moments can be considered descriptors of the image brightness surface and can be used to reconstruct an image brightness surface. Note that this method can describe the brightness surface, not just the outline of an object. However it is often applied to binary images to limit them to a description of the boundary of an object.

Moment invariants A problem with the use of raw moments as descriptors is that they are not invariant with respect to rotation, translation, and reflection of the object within the image. Hu (1962) and others have formulated functions called moment invariants which have this desired property. While useful as descriptors of an image, they have limitations. A practical problem is their sensitivity to rounding errors in the computation of the higher moments.

Average moments have been defined in two ways. Most workers suggest dividing the above moments by μ_{00}, the total density of the image (the volume under the surface). However, Dudani et al. (1977) suggest that one divide by n, the number of nonzero pixels in the image. These two methods are equivalent only for binary images (where $f(x,y) = 1$ corresponds to a point within the object and 0 otherwise). Yin and Mack (1981, p. 138) say that the latter method gives weak intensity invariance. An obvious property of central moments is that their values are invariant to translation of the object along the coordinate axes. In most studies the central moments are normalized in an effort to

eliminate the effects of overall "size" of the image. As it is in morphometrics, this is not as simple as it might seem at first. The most common normalization (due to Hu, 1962) is

$$\eta_{pq} = \mu_{pq} / \mu_{00}^{(p+q)/2+1}. \quad (32)$$

for all p, q such that $p+q = 2, 3, \ldots$. This adjusts the moments to take into account the overall intensity of the image (i.e., the volume under the brightness surface). Other normalizations are described below. Because of a misprint in Hu (1962) the divisor is often, incorrectly, given as $\mu_{00}^{(p+q)/2} + 1$. Maitra (1979, p. 697) gives the correct formula (which is confirmed by Casasent et al., 1981, p. 127).

The above moments are not useful for most studies since the coefficients are still affected by such things as rotation of the image and its degree of contrast. Several methods have been proposed for obtaining functions that are invariant to such details about an image and thus are expected to describe just its form. Hu (1962) proposed a set of absolute orthogonal invariants, h_1, (based on the normalized moments, η_{pq}, above):

$$\begin{aligned}
h_1 &= \eta_{20} + \eta_{02} \\
h_2 &= (\eta_{20}-\eta_{02})^2 + 4\eta_{11} \\
h_3 &= (\eta_{30}-3\eta_{12})^2 + (3\eta_{21}-\eta_{03})^2 \\
h_4 &= (\eta_{30}+\eta_{12})^2(\eta_{21}+\eta_{03})^2 \\
h_5 &= (\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2] \\
&\quad + (3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{21})^2-(\eta_{21}+\eta_{03})^2] \\
h_6 &= (\eta_{20}-\eta_{02})[(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2] \\
&\quad + 4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03})
\end{aligned} \quad (33)$$

and a skew invariant:

$$\begin{aligned}
h_7 &= (3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2] \\
&\quad -(\eta_{30}-3\eta_{12})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]
\end{aligned} \quad (34)$$

In addition to the position and mass invariance of the μ_{pq}, these functions are invariant to image rotation. They have been used in many applied studies. Hall (1979, p. 423) suggested the use of the logarithms of the h_i in order to reduce their "dynamic range." He does not state, however, what one should do when the $h_i \leq 0$ (which is often the case). As pointed out by Maitra (1979), absolute orthogonal invariants are sensitive to discretization and so are not computationally invariant, especially if one uses outline images. The example given by Hall (1979, p. 423) shows that they may vary over several orders of magnitude.

For practical computation, the formulas for the h_i can be simplified as follows:

$$\begin{aligned}
h_1 &= \eta_{20} + \eta_{02} \\
h_2 &= A^2 + 4\eta_{11}^2 \\
h_3 &= B^2 + C^2 \\
h_4 &= D^2 + E^2 \\
h_5 &= BF + CG \\
h_6 &= A(D^2-E^2) + 4\eta_{11}DE \\
h_7 &= CF-BG,
\end{aligned} \quad (35)$$

where

$$\begin{aligned}
A &= \eta_{20}-\eta_{02} \\
B &= \eta_{30}-3\eta_{12} \\
C &= 3\eta_{21}-\eta_{03} \\
D &= \eta_{30} + \eta_{12} \\
E &= \eta_{21} + \eta_{03} \\
F &= D(D^2-3E^2) \\
G &= E(3D^2-E^2).
\end{aligned} \quad (36)$$

Dudani et al. (1977) proposed that Hu's (1962) coefficients should be normalized to correct for differences in the scale of an image. Since magnification (isotropic scale change in both x and y-coordinates) yields an equivalent image, descriptor functions should be insensitive to such a transformation. By dividing Hu's coefficients by various

powers of h_1, the normalizations below achieve this scale invariance.

$$\begin{aligned} d_1 &= h_2/h_1^2 \\ d_2 &= h_3/h_1^3 \\ d_3 &= h_4/h_1^3 \\ d_4 &= h_5/h_1^6 \\ d_5 &= h_6/h_1^4 \\ d_6 &= h_7/h_1^6. \end{aligned} \tag{37}$$

Yin and Mack (1981, p.138) proposed a similar normalization but raised the moments to various fractional powers. The resulting moments are then in a more convenient numerical range.

$$\begin{aligned} y_1 &= \begin{cases} 1 & \text{(intensity)} \\ h_1/\mu_{00} & \text{(silhouette)} \end{cases} \\ y_2 &= \sqrt{d_1} \\ y_3 &= \sqrt[3]{d_2} \\ y_4 &= \sqrt[3]{d_3} \\ y_5 &= |\, d_4 \,|^{1/4} \\ y_6 &= |\, d_5 \,|^{1/4} \\ y_7 &= |\, d_6 \,|^{1/6}, \end{aligned} \tag{38}$$

where an intensity image is one in which the $f(x,y)$ are equal to the actual image brightness values. In a silhouette image, all brightness equal to or larger than a specified threshold have been set to 1 and all values less than the threshold set to 0. Since the d_i and the y_i are scale invariant, the normalization of the μ_{pq} by division by $\mu_{00}^{(p+q)/2+1}$ has no effect.

Reddi (1981) proposed slightly different adjustments to h_2 to h_7 which were also intended to yield scale invariant functions.

$$\begin{aligned} r_2 &= h_2/h_1^2 \\ r_3 &= h_3/h_1^{2.5} \\ r_4 &= h_4/h_1^{2.5} \\ r_5 &= h_5/h_1^5 \\ r_6 &= h_6/h_1^{3.5} \\ r_7 &= h_7/h_1^5. \end{aligned} \tag{39}$$

Contrast invariant moments, m_i, were proposed by Maitra (1979).

$$\begin{aligned} m_1 &= \sqrt{h_2}/h_1 \\ m_2 &= h_3\mu_{00}/(h_2h_1) \\ m_3 &= h_4/h_3 \\ m_4 &= \sqrt{h_5}/h_4 \\ m_5 &= h_6/(h_4h_1) \\ m_6 &= h_7/h_5. \end{aligned} \tag{40}$$

Maitra (1979) does not indicate what should be done when h_5 is negative (one could, arbitrarily, use $-\sqrt{|\, h_5 \,|}$).

The problem of the normalization of the moment invariants is more complex than one might at first expect since the various adjustments described above can interact. In a discrete image, multiplication of the x and y-coordinates by a constant effects a scale change but no change in the numbers of rows and columns in an image, and hence no change in the "mass" of an image. On the other hand, a magnification of the original image implies that the digitized image is spread out over more pixels and thus the digitized image has a larger mass. Hu's (1962) normalization compensates by, in effect, reducing the image intensities so that the mass stays the same. But this lowers the contrast of the image. Radial and angular moment invariants were proposed by Reddi (1981). He

expressed h_1 to h_7 in terms of angular and radial moments. While these are mathematically equivalent to the xy-invariants described above, Reddi (1981) showed how the polar form allows one to generalize to higher order moment invariants more easily.

White and Prentice (1987) compared the effectiveness of moment invariants, chain-code descriptors, Elliptic Fourier coefficients, and conventional measurements to discriminate between *a priori* defined groups. They found the chain-code descriptors to perform poorly and both the moments and the Fourier coefficients to perform well in their tests. However, Rohlf and Ferson (unpublished) found the method of moments to perform poorly, due in part to dependencies among some of the coefficients (they are not statistically independent) and sensitivity to rounding errors.

Use of moments to determine orientation Since the ordinary moments are sensitive to the location and orientation of an object within an image, this information can be used to determine an object's location and orientation so one can move the object into a standard position for further processing. The first eigenvector of the variance-covariance matrix gives the direction of greatest variation. If one is working with an elongated object (such as a mosquito wing), then the vector is parallel to its long axis. Using the notation of Box 15.5 of Sokal and Rohlf (1981), the slope of this line is

$$b = \frac{s_{12}}{\lambda_1 - s_1^2}, \tag{41}$$

where

$$\lambda_1 = \frac{1}{2}(s_1^2 + s_2^2 + D), \tag{42}$$

and

$$D = \sqrt{(s_1^2 + s_2^2)^2 - 4(s_1^2 s_2^2 - s_{12}^2)}. \tag{43}$$

In terms of the notation of the previous section, $s_{12} = \mu_{11}$, $s_1^2 = \mu_{20}$, and $s_2^2 = \mu_{02}$. Knowing the slope and position, the object can be rotated and translated into a standard position for subsequent analyses.

Reconstruction of Images

This topic is important for several reasons. First, if one can reconstruct the important features of an image from a set of measured parameters, that demonstrates that the parameters used are sufficient to describe the image. Of course, that does not prove that any of the parameters are directly interpretable biologically. One may have to perform various transformations on the parameters in order to put them into a form suitable for analysis and interpretation. The discussion on moment invariants, above, shows that different assumptions can suggest different transformations of the initial set of raw moments of an image surface. Reconstructed images may also be useful in themselves as convenient checks on whether the measurements are mutually consistent. If one measurement or more is inaccurate, the reconstructed image should look distorted. Strauss and Bookstein (1982) point this out as one of the advantages of the truss method.

Summary statistics such as means, confidence regions, and principal component axes can be expressed in terms of the input variables. Fourier coefficients, for example, can be averaged to give a description of an average outline. These coefficients can then be used to construct a plot of the average outline. Points within a multivariate confidence region correspond to particular combinations of values of the input parameters. It is possible, for example, to show a confidence region for a set of morphometric shapes by constructing examples of various extreme images that still belong to the confidence region. Rohlf and Archie (1984) show examples of reconstructions of hypothetical mosquito wings representing extremes possible along each principal component axis. Thus the

morphometrician is able to concentrate on the geometric aspects of the organisms under study without getting distracted by the large numbers of measurements, parameters, and various coefficients involved in the mathematical and statistical analyses being performed.

Acknowledgments

Dennis Slice assisted in getting software ready in time for the software demonstrations given as part of the presentation of this paper at the workshop (the IMAGE and FRACTAL-D programs for the IBM PC were shown to illustrate some of the points made in this manuscript). He and Karen Rohlf prepared the illustrations for this paper. Norma Watson provided many editorial suggestions. Their assistance is greatly appreciated. The discussion on the method of moment invariants is based on an unpublished study made in collaboration with Scott Ferson.

This work was supported in part by a grant (BSR 8306004) by the National Science Foundation. This paper is contribution number 761 from the Graduate Studies in Ecology and Evolution, State University of New York at Stony Brook.

References

Ashkar, G. P. and J. W. Modestino. 1978. The contour extraction problem with biomedical applications. Computer Graphics Image Processing, 7:331-355.

Ballard, D. H. and L. M. Brown. 1982. Machine vision. Prentice-Hall, New York. 523 pp.

Barnsley, M. F., V. Ervin, D. Hardin, and J. Lancaster. 1986. Solution of an inverse problem for fractals and other sets. Proc. Natl. Acad. Sci. USA, 83:1975-1977.

Blum, H. 1973. Biological shape and visual science (part I). J. Theor. Biol., 38:205-287.

Bookstein, F. L. 1981. Looking at mandibular growth: some new geometric methods. Pp. 83-103 *in* Craniofacial biology. Univ. Michigan Center for Human Growth and Development. (D. S. Carlson, ed.) Ann Arbor.

Bookstein, F. L., B. Chernoff, R. L Elder, J. M. Humphries, Jr., G. R. Smith, and R. E. Strauss. 1985. Morphometrics in evolutionary biology. The Academy of Natural Science Philadelphia. Special Publ. No. 15, 277 pp.

Bookstein, F. L., R. E. Strauss, J. M. Humphries, B. Chernoff, R. L. Elder, and G. R. Smith. 1982. A comment on the uses of Fourier methods in systematics Syst. Zool., 31:85-92.

Brice, C. R. and C. L. Fennema. 1970. Scene analysis using regions. Artificial Intell., 1: 205-226.

Casasent, D., J. Pauly, and D. Fetterly. 1981. Infrared ship classification using a new moment pattern recognition concept. SPIE, 302:126-133.

Castleman, R. R. 1979. Digital image processing. Prentice-Hall, Englewood Cliffs, New Jersey, 429 pp.

Cheung, E. and J. Vlcek. 1986. Fractal analysis of leaf shapes. Can. J. For. Res., 16:124-127.

Dudani, S. A., K. J. Breeding, and R. B. McGhee. 1977. Aircraft identification by moment invariants. IEEE Trans. Computers, C26:39-46.

Dyer, C. R. and A. Rosenfeld. 1979. Thinning algorithms for gray-scale pictures. IEEE Trans. Pattern Anal. Mach. Intell., PAMI-1:88-89.

Engles, H. 1986. A least squares method for estimation of Bezier curves and surface and its applicability to multivariate analysis. Math. Biosci., 79:155-170.

Evans, D. G., P. N. Schweitzer, and M. S. Hanna. 1985. Parametric cubic splines and geological shape descriptions. Math. Geology, 17:611-624

Ferson, S., F. J. Rohlf, and R. K. Koehn. 1985. Measuring shape variation of two-dimensional outlines. Syst. Zool., 34:59-68.

Frei, W. 1977. Image enhancement by histogram hyperbolization. Comp. Graphics and Image Processing, 6:286-294.

Grainger, J. E. 1981. A quantitative analysis of photometric data from aerial photographs for vegetation survey. Vegetatio, 48:71-82

Groch, W. D. 1982. Extraction of line shaped objects from aerial images using a special opera-

tor to analyze the profiles of functions. Computer Graphics Image Processing, 18:347-358

Hall, E. L. 1979. Computer Image Processing and Recognition Academic Press, New York, 584 pp.

Horn, B. K. P. 1986. Robot vision. M.I.T. Press, Cambridge, MA, 509 pp.

Hu, M. K. 1962. Visual pattern recognition by moment invariants. IRE Trans. Information Th., 8:179-187.

Hummel, R. 1977. Image enhancement by histogram transformation. Computer Graphics and Image Processing, 6:184-195.

Katz, M. J. and E. B. George. 1985. Fractals and the analysis of growth paths. Bull. Math. Biol., 47:273-286.

Kaesler, R. L. and J. A. Waters. 1972. Fourier analysis of the ostracode margin. Geol. Soc. Amer. Bull., 83:1169-1178.

Kirsch, R. A. 1971. Computer determination of the constituent structure of biological images. Computers and Biomedical Res., 4:315-328.

Kuhl, F. P. and C. R. Giardina. 1982. Elliptic Fourier features of a closed contour Computer Graphics and Image Processing, 18:236-258.

Lohmann, G. P. 1983. Eigenshape analysis of microfossiles: a general morphometric procedure for describing changes in shape. Math. Geol., 15:659-672.

Long, C. A. 1985. Intricate sutures as fractal curves. J. Morph., 185:285-295.

Machuca, R. and A. L. Gilbert. 1981. Finding edges in noisy scenes. IEEE Trans. Pattern Anal. Mach. Intell., PAMI-3:103-111.

Maitra, S. 1979. Moment invariants. Proc. of the IEEE., 67:697-699.

Mero, L. and Z. Vassy. 1975. A simplified and fast version of the Hueckel operator for finding optimal edges in pictures. Proc. 4th. Intl. Conf. on Artificial Intelligence. Tbilisi, USSR. 650-655.

Morse, D. R., J. H Lawton, M. M. Dodson, and M. H. Williamson. 1985. Fractal dimension of vegetation and the distribution of arthropod body lengths. Nature, 314:731-733.

Nagao, M. and T. Matsuyama. 1979. Edge preserving smoothing. Computer Graphics Image Processing, 9:394-407.

Papoulis, A. 1965. Probability, random variables, and stochastic variables. McGraw-Hill, New York.

Paton, K. 1979. Line detection by local methods. Computer Graphics Image Processing, 9: 316-332.

Pavlidis, T. 1982. Algorithms for graphics and image processing. Computer Science Press: Rockville, MD, 416 pp.

Peli, T. and J. S. Lim. 1982. Adaptive filtering for image enhancement. Optical Engineering, 21: 108-112.

Reddi, S. S. 1981. Radial and angular moment invariants for image identification. EEE Trans. Pattern Anal. Mach. Intell., PAMI-3:240-242.

Riseman, E. M. and M. A. Arbib. 1977. Computational techniques in visual systems, part II: segmenting static scenes. IEEE Computer Soc. Repository, R:77-87

Rohlf, F. J. and J. Archie. 1984. A comparison of Fourier methods for the description of wing shape in mosquitoes (Diptera: Culicidae). Syst. Zool., 33:302-317.

Rohlf, F. J. and S. Ferson. 1983. Image analysis. Pp. 583-599. *in* Numerical taxonomy. (J. Felsenstein, ed.) Springer-Verlag, Berlin

Rosenfield, A. and A. C. Kak. 1982. Digital picture processing, volume 1. Academic Press, New York, 435 pp.

Samet, H. 1981a. An algorithm for converting rasters to quadtrees. IEEE Trans. Pattern Anal. Mach. Intell., PAMI-3:93-95

Samet, H. 1981b. Computing perimeters of regions in images represented by quadtrees. IEEE Trans. Pattern Anal. Mach. Intell., PAMI-3: 683-687.

Samet, H. 1983. A quadtree medial axis transform. Communications ACM, 26:680-393.

Scott, D. S. and S. Iyengar. 1986. TID--A translation invariant data structure for storing images. Comm. ACM., 29:418-429.

Shaw, G. B. 1979. Local and regional edge detectors: some comparisons. Computer Graphics and Image Processing, 9:135-149

Sokal, R. R. and F. J. Rohlf. 1981. Biometry. Freeman, New York, 859 pp.

Strauss, R. E. and F. L. Bookstein. 1982. The truss: body form reconstructions in morphometrics. Syst. Zool., 31:113-135.

Tanimoto, S. and T. Pavlidis. 1975. A hierarchical data structure for picture processing. Computer Graphics Image Proc., 4:104-119.

Vlcek, J. and E. Cheng. 1986. Fractal analysis of leaf shape. Can. J. Forestry Res., 16:124-127.

Wang, S., A. Y. Wu, and A. Rosenfeld. 1981. Image approximation from gray scale "medial axes." IEEE Trans. Pattern Anal. Mach. Intell, PAMI-3:687-??.

White, R. J. and H. C. Prentice. 1987. Comparison of shape description methods for biological outlines. Pp. 395-402 *in* Classification and related methods of data analysis. (H. H. Bock, ed.). North-Holand, New York.

Yin, B. H. and H. Mack. 1981. Target classification algorithms for video and forward looking infrared (FLIR) imagery. SPIE, Infrared Technology for Target Detection and Classification, 302:134-140.

Younker, J. L. and R. Ehrlich. 1977. Fourier biometrics: harmonic amplitudes as multivariate shape descriptors. Syst. Zool., 26:336-342.

Zahn, C. T. and R. Z. Roskies. 1972. Fourier descriptors plane closed curves. IEEE Trans. on Computers, 21:269-281.