Lecture 9: Shape Description (Regions)

©Bryan S. Morse, Brigham Young University, 1998–2000
Last modified on February 16, 2000 at 4:00 PM

Contents

Reading

SH&B, 6.3.1-6.3.3
Castleman 19.1-19.2.1.1; 19.3.1-19.3.3.3

9.1 What Are Descriptors?

In general, descriptors are some set of numbers that are produced to describe a given shape. The shape may not be entirely reconstructable from the descriptors, but the descriptors for different shapes should be different enough that the shapes can be discriminated.

We saw some simple examples of descriptors in our discussion of contour representations. These included the contour length (perimeter) and the bending energy.

What qualifies as a good descriptor? In general, the better the descriptor is, the greater the difference in the descriptors of significantly different shapes and the lesser the difference for similar shapes. What then qualifies similarity of shape? Well, nobody's really been able to answer that one yet. If we could quantify similarity of shape, we'd have the perfect descriptor. Indeed, that's what descriptors are: attempts to quantify shape in ways that agree with human intuition (or task-specific requirements).

Regions can either describe boundary-based properties of an object or they can describe region-based properties. In this lecture, we focus on region-based descriptors.

9.2 Some Simple Shape Descriptors

A few simple descriptors are

Area. The number of pixels in the shape. Your text describes algorithms for calculating the area from quadtree or chain-coding representations.

Perimeter. The number of pixels in the boundary of the shape.

(Non-)Compactness or (Non-)Circularity. How closely-packed the shape is (not): perimeter^2/area. The most compact shape is a circle (4π). All other shapes have a compactness larger than 4π.

Eccentricity. The ratio of the length of the longest chord of the shape to the longest chord perpendicular to it. (This is one way to define it—there are others.)

Elongation. The ratio of the height and width of a rotated minimal bounding box. In other words, rotate a rectangle so that it is the smallest rectangle in which the shape fits. Then compare its height to its width.

Rectangularity. How rectangular a shape is (how much it fills its minimal bounding box): area of object/area of bounding box. This value has a value of 1 for a rectangle and can, in the limit, approach 0 (picture a thin X).

Orientation. The overall direction of the shape. (I'll come back and define this more precisely later.)

9.3 Practical Considerations for Measuring Properties

When measuring the perimeter of an object, do we follow the pixels on the inside of the object's boundary, the exterior pixels just outside that boundary, or the "cracks" between the interior and exterior pixels? Any of the three methods can be used, and the differences go away as the object size increases—just be consistent.

Also, when measuring perimeter, does one simply count pixels? That can artificially bias diagonal edges high (if counting 4-connected) or low (if counting 8-connected). We can do what we discussed earlier for distance measures: count 1 pixel for 4-connected neighbors and $\sqrt{2}$ for 8-connected ones.

9.4 Topological Descriptors

If we stretch a shape like we would a cutout from a rubber sheet, there are certain shapes we can make and others we can't. *Topology* refers to properties of the shape that don't change, so long as you aren't allowed to tear or join parts of the shape.

A useful topological descriptor is the *Euler number* E: the number of connected components C minus the number of holes H:

$$E = C - H$$

(You should know how to use connected-component labeling to determine the Euler number.) While the Euler number may seem like a simple descriptor (it is), it can be useful for separating simple shapes.

9.5 Convex Hull: Bays

Although not strictly a topological property, we can also describe shape properties by measuring the number or size of concavities in the shape. We can do this first finding the convex hull of the shape and then subtracting the shape itself. What are left are either holes ("lakes") or concavities ("bays"). This might be useful in trying to distinguish the letter "O" from "C", etc. One can then recursively identify concavities within each "bay", etc. The resulting structure is a *concavity tree*.

9.6 Extremal Points

Another way to describe a shape is to find its *extremal points*.

The simplest form of extremal representation is the *bounding box*: the smallest rectangle that completely contains the object.

A more powerful way of using extremal points is to find the eight points defined by: top left, top right, left top, left bottom, bottom right, bottom left, right top, and right bottom. (Obviously, some of these points may be the same.)

By connecting opposing pairs of extremal points (top left to bottom right, etc.) we can create four axis that describe the shape. These axis can themselves be used as descriptors, or we can use combinations of them.

For example, the longest axis and its opposite (though not necessarily orthogonal axis) can be labeled the *major and minor axes*. The ratio of these can be used to define an *aspect ratio* or *eccentricity* for the object. The direction of the major axis can be used as an (approximate) orientation for the object.

How do extremal points respond to transformations?

9.7 Profiles

A useful region-based signature is the *profile* or *projection*. The *vertical profile* is the number of pixels in the region in each column. The *horizontal profile* is the number of pixels in the region in each row. One can also define *diagonal profiles*, which count the number of pixels on each diagonal.

Profiles have been used in character recognition applications, where an L and a T have very different horizontal profiles (but identical vertical profiles), or where A and H have different vertical profiles (but identical horizontal profiles).

Profiles are useful as shape signatures, or they may be used separate different regions. A technique known as "signature parsing" uses vertical profiles to separate horizontally-separated regions, then individual horizontal profiles to recursively separate vertically-separated regions, etc. By alternating this process and recursively splitting each region, one "parse" binary objects that are organized horizontally and vertically, like text on a page.

9.8 Moments

Another way to describe shape uses statistical properties called *moments*.

9.8.1 What Statistical Moments Are

For a discrete one-dimensional function $f(x)$, we can compute the mean value of the function using

$$\mu = \frac{\sum_{x=1}^{N} x f(x)}{\sum_{x=1}^{N} f(x)} \tag{9.1}$$

We can also describe the variance by

$$\sigma^2 = \frac{\sum_{x=1}^{N} (x-\mu)^2 f(x)}{\sum_{x=1}^{N} f(x)} \tag{9.2}$$

A third statistical property, called *skew*, describes how symmetric the function is:

$$\text{skew} = \frac{\sum_{x=1}^{N} (x-\mu)^3 f(x)}{\sum_{x=1}^{N} f(x)} \tag{9.3}$$

All of these are examples of moments of a function.

One can define moments about some arbitrary point, usually either about zero or about the mean. The n-th moment about zero, denoted as m_n, is

$$m_n = \frac{\sum_{x=1}^{N} x^n f(x)}{\sum_{x=1}^{N} f(x)} \tag{9.4}$$

The zeroeth moment, m_0, is always equal to 1. The mean μ is the first moment about zero:

$$\mu = m_1 \tag{9.5}$$

The n-th moment about the mean, denoted as μ_n and called the n-th *central moment* is

$$\mu_n = \frac{\sum_{x=1}^{N} (x-\mu)^n f(x)}{\sum_{x=1}^{N} f(x)} \tag{9.6}$$

The zeroeth central moment μ_0 is, again, equal to 1. The first central moment μ_1 is always 0. (Do you see why?) The second central moment μ_2 is the *variance*:

$$\sigma^2 = \mu_2 \tag{9.7}$$

The third central moment μ_3 is the *skew*:

$$\text{skew} = \mu_3 \tag{9.8}$$

The fourth central moment μ_4 is the *kirtosis*:

$$\text{kirtosis} = \mu_4 \tag{9.9}$$

If we have an infinite number of central moments, we can completely describe the shape of the function. (We do, of course, also need the mean itself to position the function).

9.8.2 Moments of Two-Dimensional Functions

Suppose that we have a discrete function not over one variable but of two. We can extend the concept of moments by defining the ijth moment about zero as

$$m_{ij} = \frac{\sum_{x=1}^{N}\sum_{y=1}^{N} x^i y^j f(x,y)}{\sum_{x=1}^{N}\sum_{y=1}^{N} f(x,y)} \tag{9.10}$$

Again, $m_{00} = 1$. The m_{10} is the x component μ_x of the mean and m_{01} is the y component μ_y of the mean.

We define the *central moments* as

$$\mu_{ij} = \frac{\sum_{x=1}^{N}\sum_{y=1}^{N} (x-\mu_x)^i (y-\mu_y)^j f(x,y)}{\sum_{x=1}^{N}\sum_{y=1}^{N} f(x,y)} \tag{9.11}$$

Similar to before, $\mu_{10} = \mu_{01} = 0$.

We can use these moments to provide useful descriptors of shape. Suppose that for a binary shape we let the pixels outside the shape have value 0 and the pixels inside the shape value 1. The moments μ_{20} and μ_{02} are thus the variances of x and y respectively. The moment μ_{11} is the *covariance* between x and y, something you may have already seen in other courses. You can use the covariance to determine the orientation of the shape. The covariance matrix C is

$$C = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \tag{9.12}$$

By finding the eigenvalues and eigenvectors of C, we can determine the eccentricity of the shape (how elongated it is) by looking at the ratio of the eigenvalues, and we can determine the direction of elongation by using the direction of the eigenvector with whose corresponding eigenvalue has the largest absolute value. The orientation can also be calculated using

$$\theta = \frac{1}{2}\tan^{-1}\frac{2\mu_{11}}{\mu_{20}-\mu_{02}} \tag{9.13}$$

(Note: be careful when implementing this—there is a 180-degree ambiguity in arctangents that turns into a 90-degree ambiguity when you take half the angle. If you're coding this in C or Java, use `atan2` instead of `atan`.

There are many other useful ways that we can combine moments to describe a shape. Even more fundamentally, though, we can completely reconstruct the shape if we have enough moments. But what if we don't use enough? Well, we can construct approximations to the shape that share these same properties. I could, for example, make a circle with the same mean as the shape, an ellipse with the same mean and covariance, etc.

In a sense, the sequence of moments is analogous to the components of a Fourier sequence—the first few terms give the general shape, and the later terms fill in finer detail.

Moments have been shown to be a very useful set of descriptors for matching.

9.8.3 Profile Moments

A number of satistical moments for 2-d regions can be calculated from the profiles of those regions. Let's define the following profiles:

$$\begin{aligned} P_v &= \text{vertical profile of } f(x,y) \\ P_h &= \text{horizontal profile of } f(x,y) \\ P_d &= \text{diagonal (}45\text{ degree) profile of } f(x,y) \\ P_e &= \text{diagonal (}-45\text{ degree) profile of } f(x,y) \end{aligned}$$

$$\begin{aligned}\mu_{10} &= \mu_v \\ \mu_{01} &= \mu_h\end{aligned}$$

$$\begin{aligned}\mu_{20} &= \mu_{vv} \\ \mu_{02} &= \mu_{hh} \\ \mu_{11} &= \frac{1}{2}\left(\mu_{dd} - \mu_{hh} - \mu_{vv}\right)\end{aligned}$$

These can then be used to derive the object orientation. This way of computing orientation can be much more efficient than calculating the 2-dimensional moments directly.

9.8.4 Response to Transformations

Finally, let's consider how moments respond to transformations.

Translation. If we translate the object, we only change the mean, not the variance or higher-order moments. So, none of the central moments affected by translation.

Rotation. If we rotate the shape we change the relative variances and higher-order moments, but certain quantities such as the eigenvalues of the covariance matrix are invariant to rotation.

Scaling Resizing the object by a factor of S is the same as scaling the x and y coordinates by S. This merely scales $x - \mu_x$ and $y - \mu_y$ by S as well. Hence, the n-th moments scale by the corresponding power of S^n. Ratios of same-order moments, such as the ratio of the eigenvalues of the covariance matrix, stay the same under scaling, as do area-normalized second-order moments.

By combining moments, we can thus produce *invariant moments*, ones that are invariant to rotation, translation, and scale. The methods for doing so for second-order moments are straightforward. The methods for doing so for higher-order moments is more complicated but also possible.

Vocabulary

- Invariance
- Euler number
- Concavity Tree
- Compactness
- Eccentricity
- Orientation
- Profile
- Statistical Moments
- Central Moment
- Variance
- Skew

- Kirtosis
- Covariance
- Covariance Matrix