9. PROPERTIES OF IMAGE OBJECTS

Objects detected in images are usually represented as either
boundaries - 1D structures (lines, open curves, closed curves), or
regions - 2D structures (regions, blobs)

These representations are usually *pictorial* and as such they are not very useful for describing properties of the objects or for reasoning about them. Explicit *functional* representations of various kinds are more suitable. In this unit we are going to cover geometric, tonal, topological and textural properties of the image objects. The phrase "image objects" is used deliberately here, to stress that the properties are related to the images of the objects rather than to the objects themselves. For example, most everyday objects are 3-dimensional whereas image objects are 2-dimensional. For the reminder of this Unit we shall be referring always to the image objects.

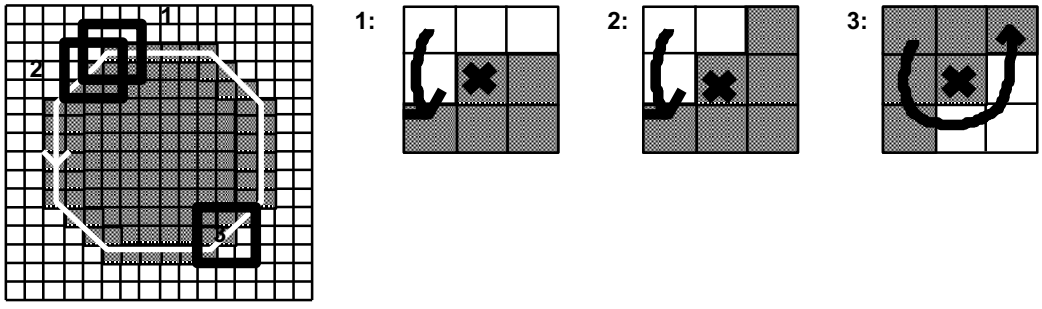
Object shape and size are the most commonly required geometric properties of the image objects. Shape description is still an active research area. This unit will describe only simplest techniques, which most commonly begin with describing the object boundary as a set of (x,y) image coordinates. This topic will be described first.

Deriving object boundaries from grey level images

The two techniques, contour following in grey level images, and graph searching, have already been encountered in the Unit 6 whilst discussing the linking of fragmented boundaries. In both techniques the coordinates of all the boundary points are stored as a boundary description. They are often processed further (see below).

Deriving object boundaries from binary images

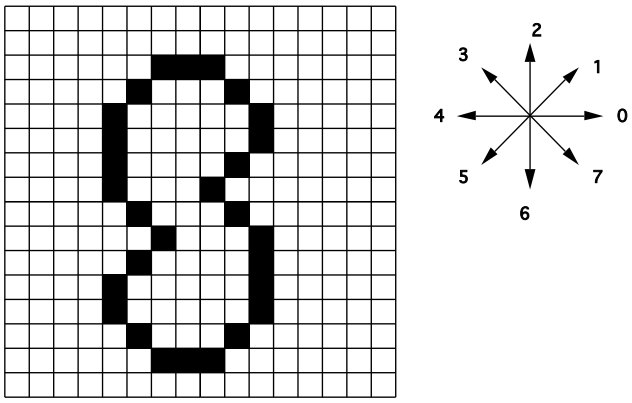
This technique (also known as *outlining*) is used to get edge coordinates of a region extracted by one of the region segmentation techniques (e.g. thresholding). No *a priori* knowledge is required about a shape. One of the simplest techniques is the "hand on the wall" approach (to be discussed in the lecture). This process produces a list containing coordinates of the region's outline.



Boundary representations

Representing the object contour by the set of (x,y) coordinates is the simplest way. This representation is usually the first step in shape description. Further, more complex representations, can provide more meaningful shape descriptions. A simple variation on the (x,y) coordinates are *chain codes*. In this representation every boundary point is stored in the

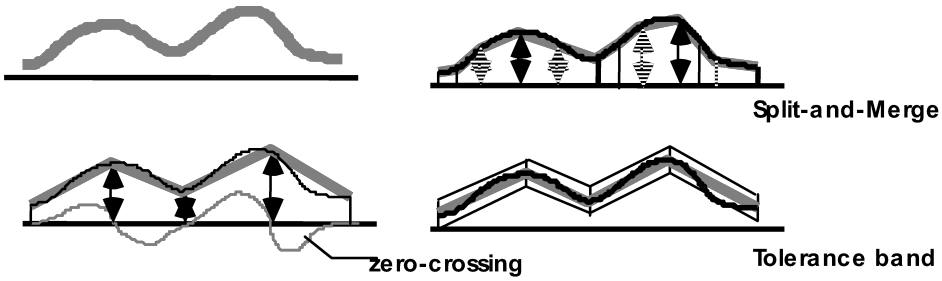
form of the *direction code*, which for each point a relative position of the next boundary point. Coordinates of the first point are stored explicitly, as an (x,y) coordinate pair.



Polylines

Polyline representation seeks to approximate the contour by a number of line segments. it consists simply of the list of endpoints. The main challenge is to find suitable locations on the contour which subdivide it into the segments. The most common techniques are

- differentiation
- split-and-merge
- tolerance bands (strip trees)



Notes:

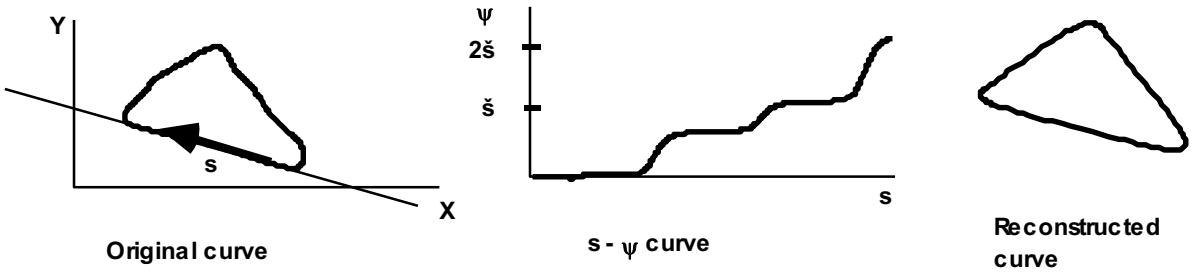
Strip trees

- A hierarchical data structure for the representation of curves.
- A binary tree where each node represents a "strip segment" - a rectangle covering a segment of the curve.
- The orientation depends on the line joining two endpoints.
- Width of the rectangle is the minimum width which gives a rectangle covering the segment.

s – ψ curves

This representation is based on the curvature of the object boundary. The algorithm builds a graph (ψ–s curve) which shows the angle between a fixed line and a line tangential to the boundary at every point on the contour. The graph has the following properties:

- horizontal lines in ψ-s space correspond to straight lines in the boundary
- non-horizontal lines correspond to segments of circles (ψ changing at a constant rate)
- curve in ψ-s segmented to yield a description of the boundary in terms of line segments and arcs



Notes:

Fourier descriptors

This representation depicts frequency characteristics of the contour. A boundary is represented as a periodic function which can be expanded in a Fourier series (a frequency domain description). The more coefficients we use to represent the curve the better approximated is the shape (See, e.g. Gonzalez & Woods, section 8.2.3).

Geometric shape descriptors (shape properties)

So far we have discussed the techniques for deriving functional representations of image object boundaries. In this section we focus on computing shape properties.

Area

This is simply the number of pixels belonging to a given image object.

Eccentricity (elongation)

x to y ratio of the *bounding rectangle*
ratio of the length of the maximum chord to a maximum chord perpendicular to it

Compactness

ratio of perimeter^2 to area, minimised by a disk

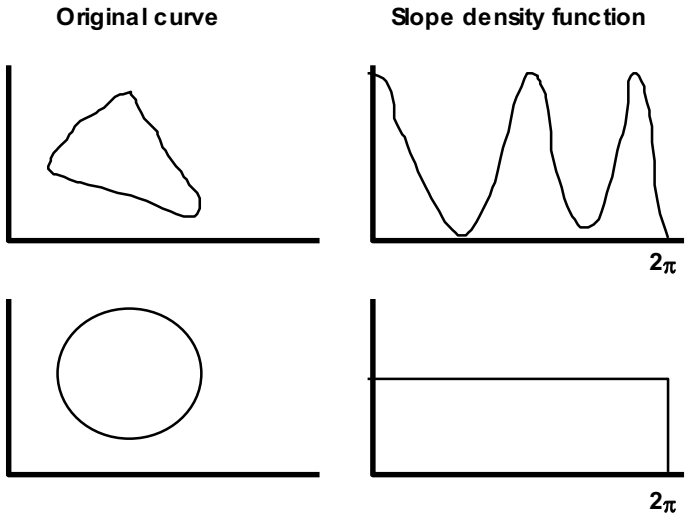
Curvature

$$|\kappa(s)|^2 = \left(\frac{d^2x}{ds^2}\right) + \left(\frac{d^2y}{ds^2}\right)$$

Notes:

Slope density function

This description is derived from a ψ–s curve; it is simply a histogram of ψ collected over the whole boundary.



Fractal dimension

Fractal dimension is a term coined by Mandelbrot. In two-dimensional space it characterises the plane-filling ability of a curve. For example a circle has fractal dimension of 1; a

geometrically constructed figure called “Triadic Koch Island” shown below has fractal dimension of 1.26; a disk has fractal dimension of 2, which indicates that it is capable of entirely filling the plane. Thus fractal dimension of a planar figure is in the range between 1.0 and 2.0. Fractal dimension is derived by using different step lengths to measure the perimeter of the figure. The estimated perimeter increases as the step length decreases. For a given step s the estimated perimeter length L can be expressed by the equation:

$$L(s) = s \ N(s)$$

where N(s) is the number of sides of length s of a polygon which approximates the perimeter. Mandelbrot made an important observation that the number of polygon sides for a given perimeter is a function of the step length and expressed this relationship as:

$$N(s) = \lambda \ s^{-D}$$

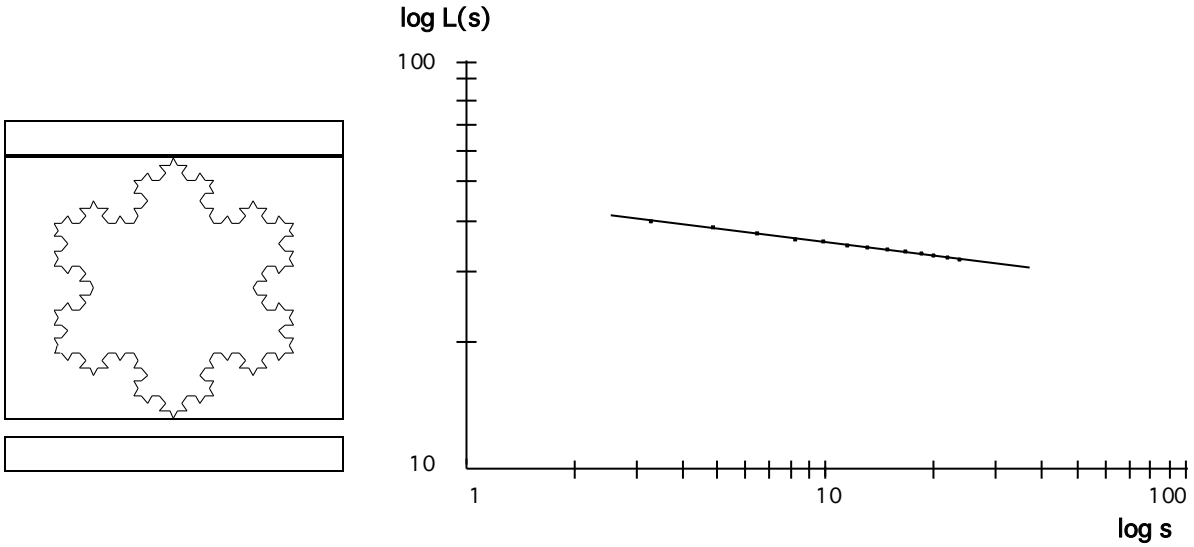
giving the final expression for the estimated perimeter length:

$$L(s) = \lambda \ s^{1 - D}$$

By taking logarithm of both sides a line equation (log L(s) vs log s) is obtained

$$\log L(s) = \log \lambda + (1 - D) \log s$$

where (1 - D) is the slope of the line and log λ is the intercept. The parameter D, which can be a fractional number, is the fractal dimension; log λ is related to the scale. Figure below shows the plot of log L(s) versus log s for a set of measurements obtained for a Triadic Koch Island.



Texture

Texture is characterised by the *tonal primitives* and the *spatial inter-relations* between them.

Tone v/s texture

- *Tonal* properties depend only on grey levels and are independent of the spatial distribution of grey levels within the area.
- *Textural* properties relate to the way the pixels with various grey levels are spatially arranged within the area and depend both on grey level value and on the spatial distribution of grey levels.
- Distribution of grey levels can be regular or irregular.

Approaches

statistical, spectral, structural and syntactic.

Statistical approaches to texture characterisation

First order statistics (mean, standard deviation) are measures of *overall* variation in contrast and therefore tell nothing about spatial distribution. Second order statistics are most frequently used. Measures derived from these statistics describe various texture attributes, such as coarseness, homogeneity and contrast.

Textural properties are expressed as the statistics of the large population of some "local" properties of the image; the local properties characterise distribution and relationships among the pixels of various grey levels. This approach is used for textures without regular, repetitive patterns, mainly for pattern recognition, i.e. to classify instances of a texture in an image into a set of classes. Some of the methods are described below.

Grey level run lengths

Grey level run length is the length of a sequence of pixels, in a particular direction, which are deemed to be uniform. This length is stored in a matrix for each uniform class and for each direction.

Co-occurrence (spatial dependency) matrix

A co-occurrence matrix is a two-dimensional matrix, indexed in each direction by grey-level values. The position (i,j) in the matrix stores the number of times that a pixel with value i has as its neighbour a pixel with value j at a given distance and for a given angle. Thus for each image a number of co-occurrence matrices can be generated. Texture measures can then be derived from these matrices, e.g.

$$\text{uniformity} = \sum_{i}^{MGL} \sum_{j}^{MGL} p(i,j)$$

$$\text{contrast} = \sum_{n=0}^{MGL-1} n^2 \Big[\sum_{i=0}^{MGL} \sum_{j=0, |i-j|=n}^{MGL} p(i,j) \Big]$$

$$\text{entropy} = -\sum_{i=0}^{MGL} \sum_{j=0, |i-j|=n}^{MGL} p(i,j) \log[p(i,j)]$$

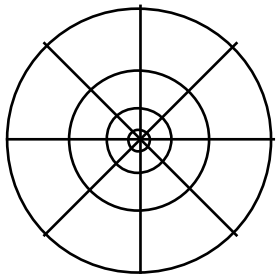
MGL - max. no. of grey levels, p(i,j) = no.of co-occurrences / total no.of co-occurrences

Spectral approaches to texture characterisation

The principle behind the spectral approach is that the power spectrum of a 2D Fourier transform of an image contains information about frequencies of grey level values in the image. If the power spectrum is divided into a sectors, as in the figure below, then the sum of

values in each sector gives a measure of both “granularity” and direction of texture: sectors near the origin will have high values for coarse texture and those further away - for fine texture.

Sums of spectrum values over the rings concentrated in origin give measures of granularity; Sums of spectrum values over sectors emanating from origin give measures of directionality.



Notes:

Further reading and exploration

Sonka, M. et al, Sections 5.2.3-5.2.5, Chapter 6 and Chapter 13.
Gonzalez, R.C. & Woods, R.E., Chapter 8.

HIPR

none

CVIP

Segment an image (e.g. Utilities -> Convert -> Binary threshold)
Label the segmented image: Utilities -> Convert -> Label image
Extract features from the labelled image:
Analysis -> Features
- Binary object features
- RST-invariant moment-based features
- Histogram features
- Spectral features