Feature description

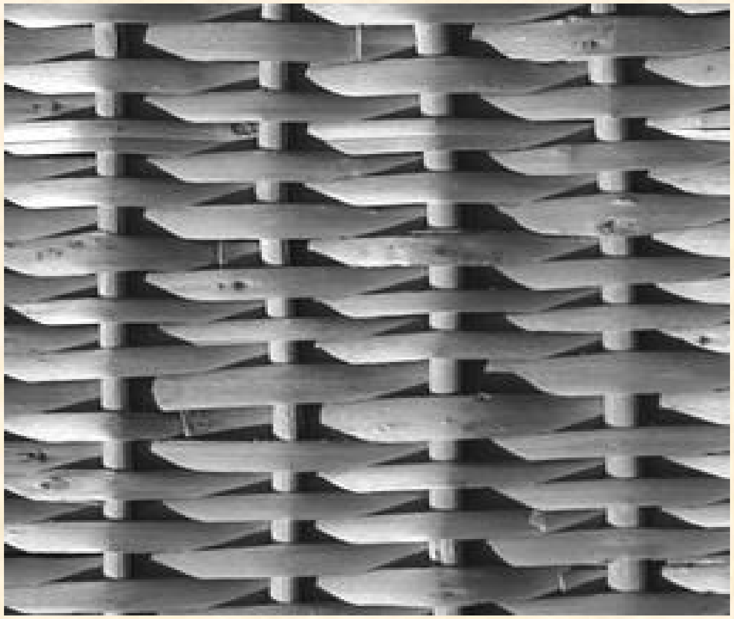
After an image has been segmented the detected region needs to be described (represented) in a form more suitable for further processing.

Representation of an image region can be carried out in two ways:

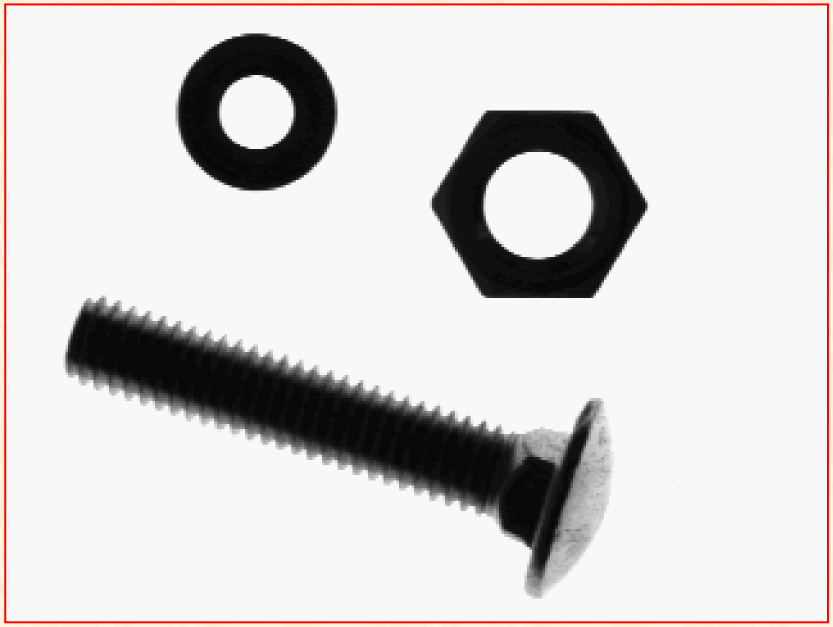
- by characterising features of its boundary, i.e. edge of a region,
- by characterising features of its interior, i.e. set of pixels constituting the region

Feature description

Region-based description



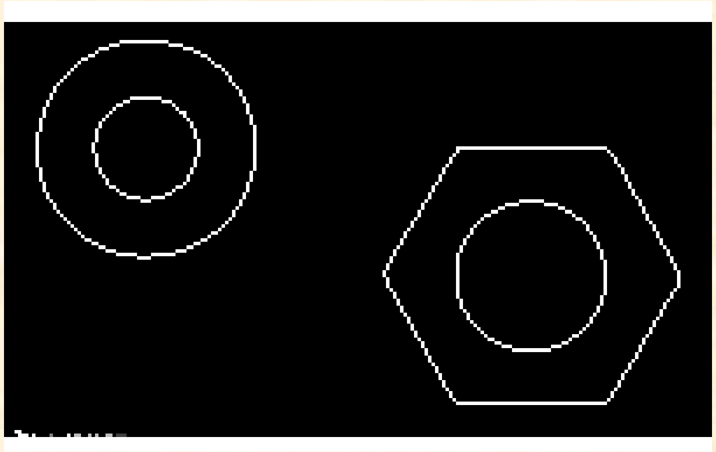
Boundary-based description



Boundary (edge) representation

Boundaries are linked edges that characterise shape of an object. They are required in computation of geometry features such as size and orientation.

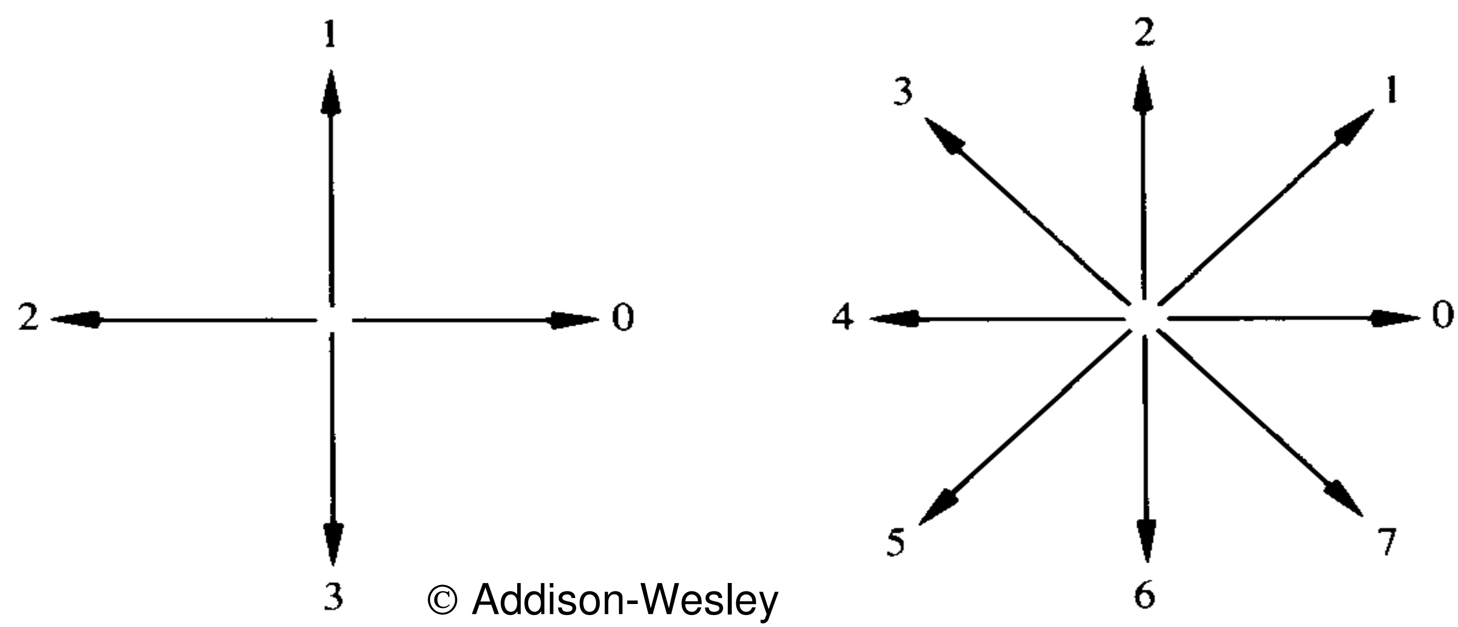
Edge representation methods aim at describing information about their shape in a way that is more suitable for coding and analysis (e.g., by means of transforming information from 2-D data to 1-D data).



Chain codes (Freeman codes)

Chain codes are used to represent a boundary by a connected sequences of straight-line segments of specified length and direction.

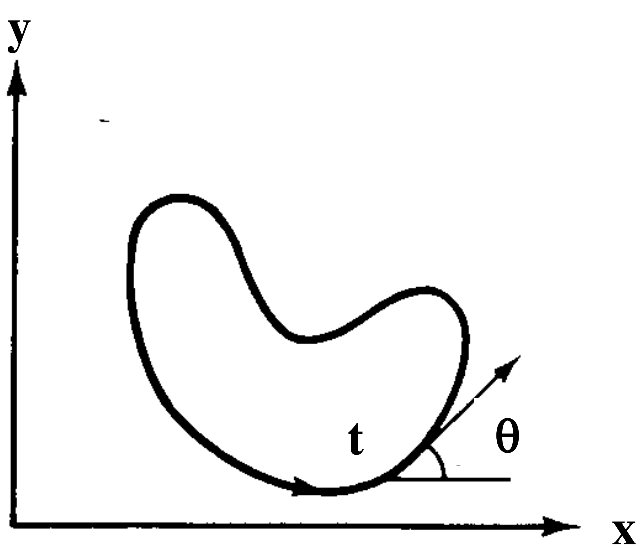
Typically, Freeman codes are based on the **four-** or **eight-connectivity** neighbourhoods.



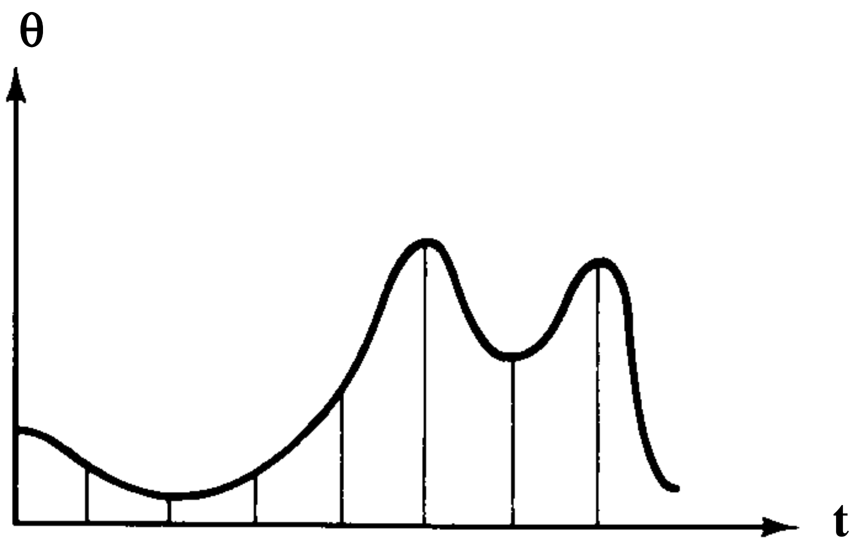
Chain codes (Freeman codes)

Chain codes can be generalised by increasing the number of allowed directions for connecting successive boundary pixels.

© Addison-Wesley



parametric representation of a contour

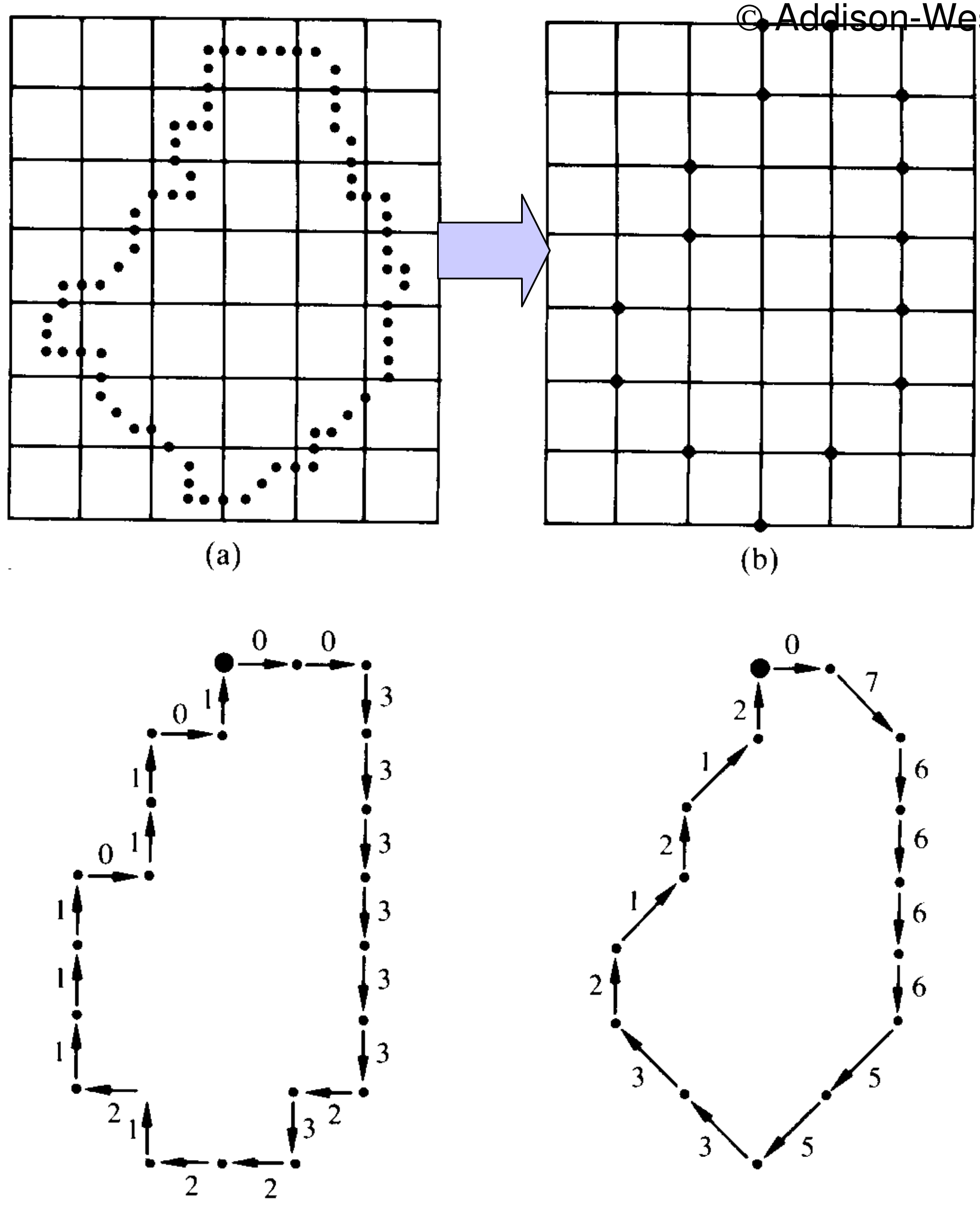


θ(t)* - angle θ vs. parameter *t

Chain codes

Small disturbances along the boundary due to noise can cause false contour values.

It can be eliminated by selecting larger grid spacing.

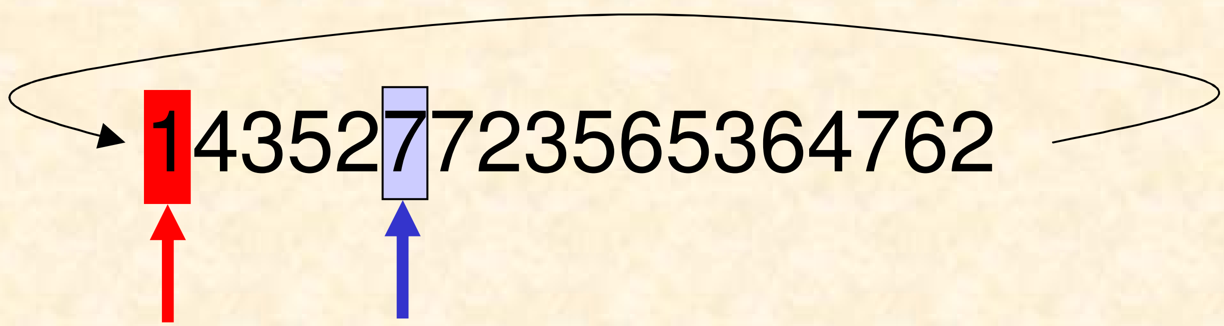


(a) (b) (c) (d)

Chain codes

The chain code of a boundary depends on the starting point. This is how the code can be normalised:

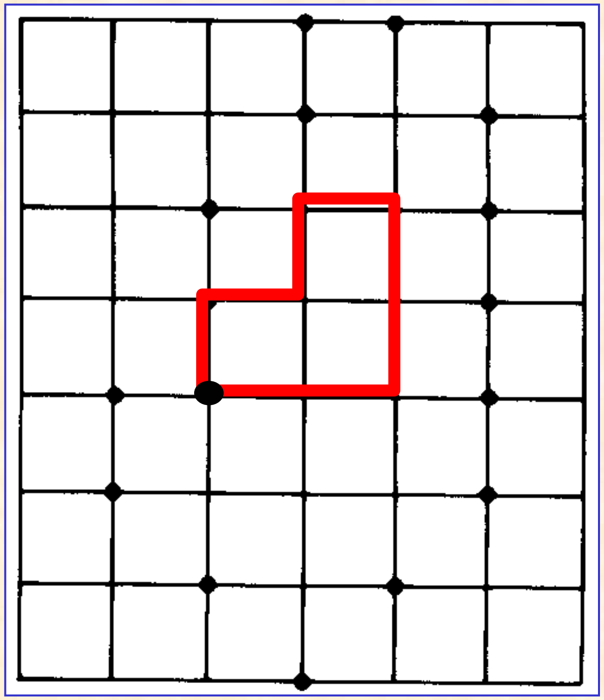
„A chain code generated by an arbitrary location is treated as an integer number, the starting point can be redefined by finding the starting point for which this number has maximum (or ***minimum****) magnitude."*



Chain codes

One can also normalise the code for rotation by using first difference of the chain code instead of the code itself (e.g., the first difference of the 4-direction code 10103322 is 3133030).

Size can be normalised by altering the size of the resampling grid.



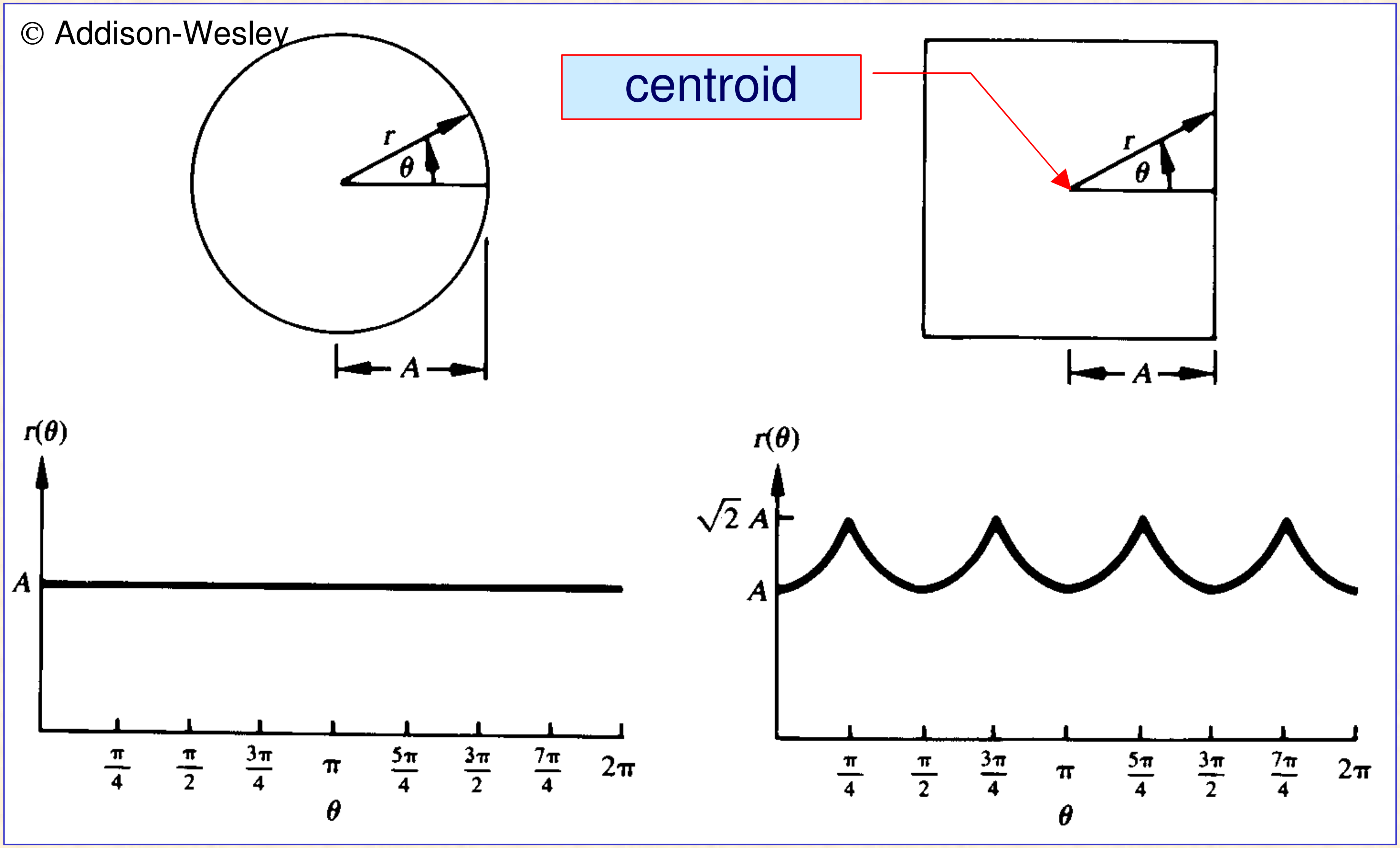
Signatures

A **signature** is a 1-D functional representation of a boundary. The simplest representation is to plot the distance from the centroid to the boundary as a function of angle.

Making signatures invariant:

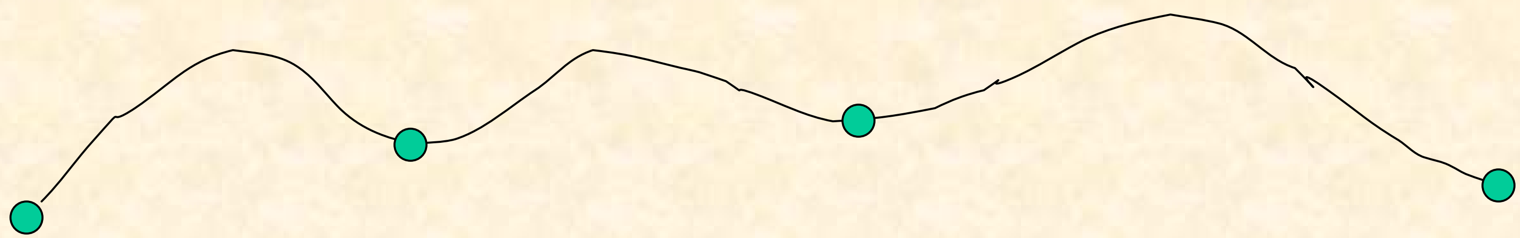
- dependence on rotation can be rejected by choosing the starting point as the point farthest from the centroid;
- normalisation for scaling can be implemented by removing the mean value of the signature function.

Geometrical figures and their signatures



B-spline Representation

B-splines are piecewise polynomial functions that can provide local approximations of contour shapes using a small number of parameters. B-splines are used in shape synthesis and analysis, computer graphics, and recognition of shapes from boundaries.



B-spline Representation

A general expression for a B-spline function is:

a) $$p(x) = p_i(x) \qquad x_i \le x \le x_{i+1} \qquad i = 0, 1, \ldots, k-1$$

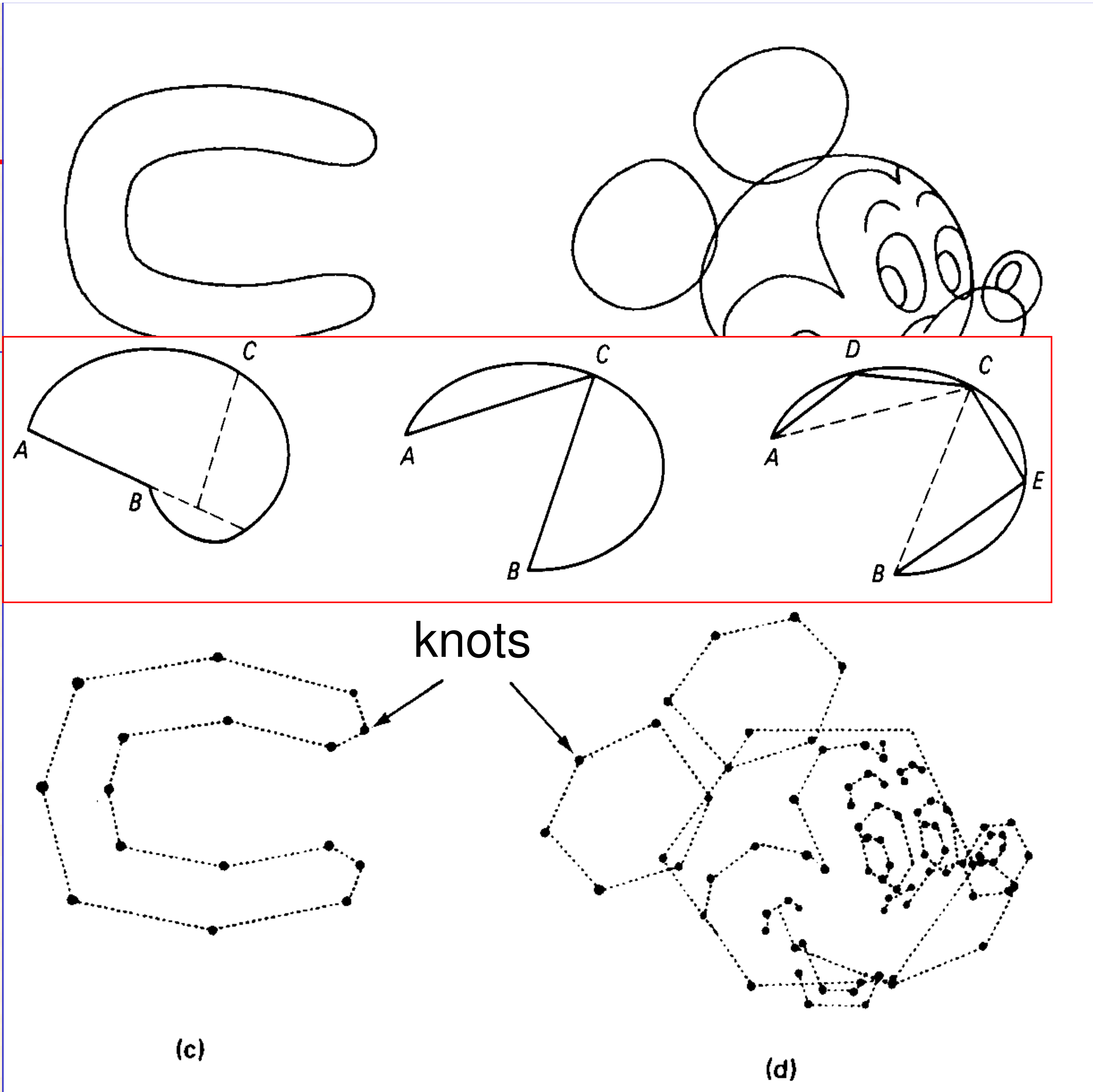
b) $$p_i^j(x_i) = p_{i+1}^j(x_i) \quad j = 0, 1, \ldots r-1, i = 1, \ldots, k-1$$

where $x_1, \ldots, x_{k-1}$ are called the breakpoints (or knots),

$p_i(x)$ are polynomials, and $p_i^j(x)$ is its *j*-th derivative.

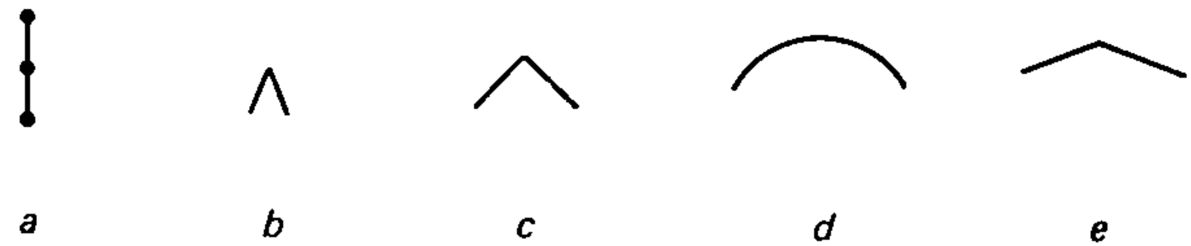
(b) requires that derivatives of the neighbouring B-splines at the knots are equal (in computer graphics applications r=3).

B-splines in contour representation

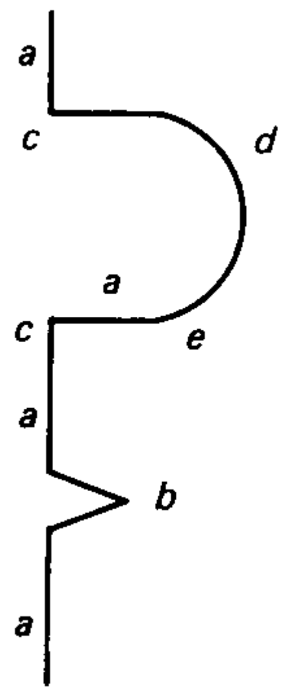


Syntactic representation

Primitive structural symbols

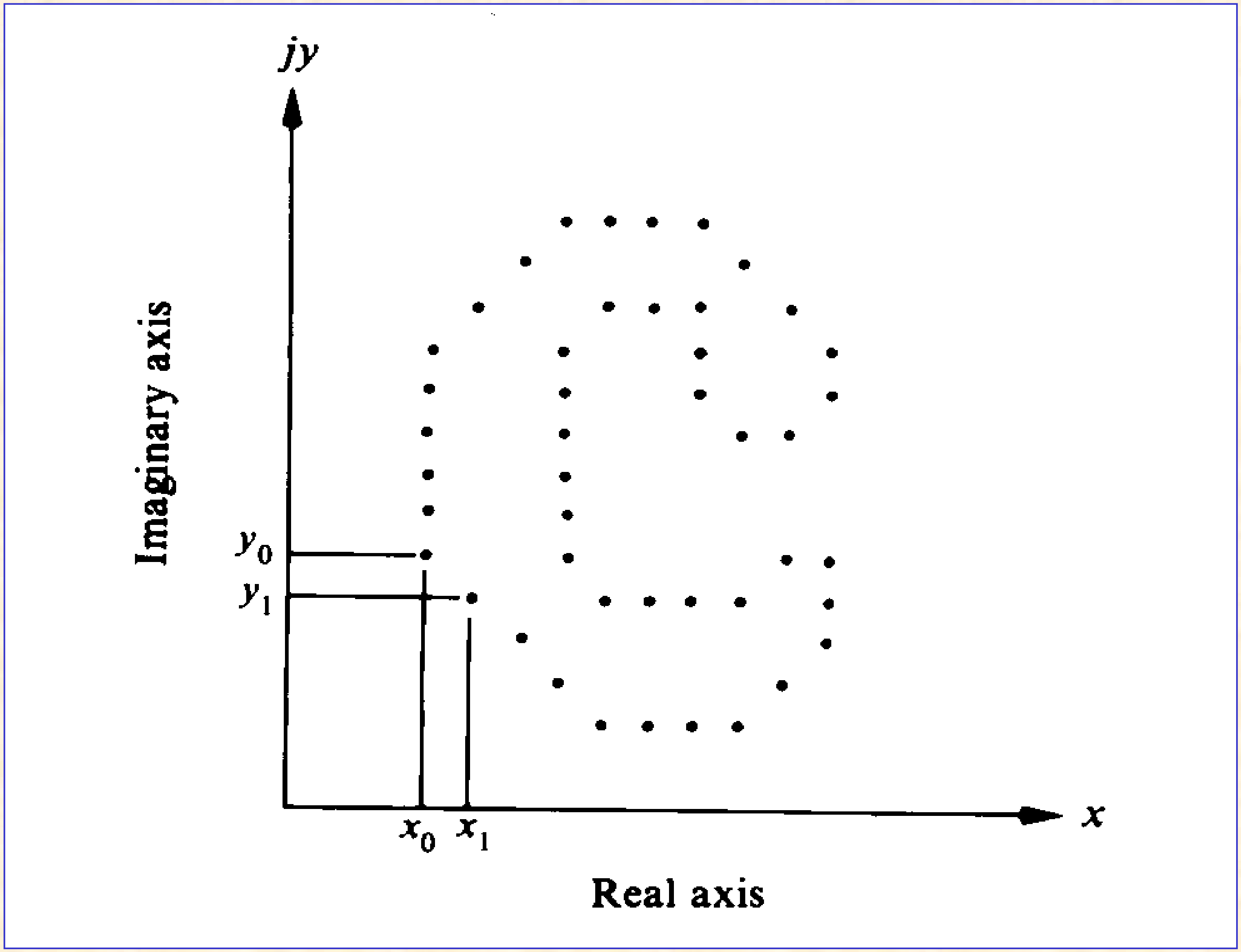


Object structure



In syntactic representation the boundary is decomposed into the set of primitive sub-pattern shapes. By adding a syntax, such as connectivity rules, it is possible to obtain a syntactic representation, which is simply a string of symbols, each representing a *primitive*.

Fourier Descriptors



Fourier Descriptors

Once the boundary trace is known, it can be considered as a pair of waveforms x(t), y(t):

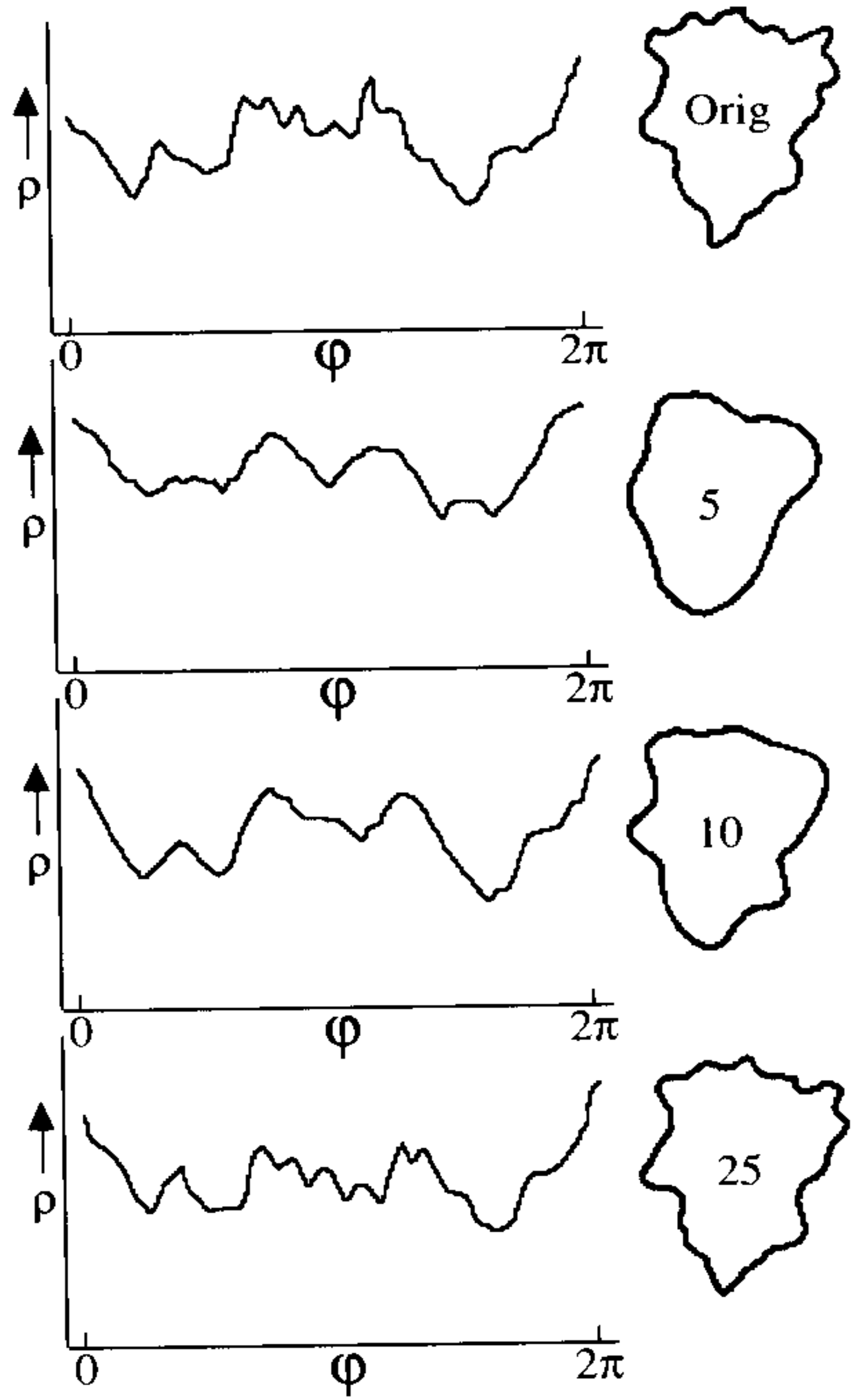
$$s(k) = x(k) + jy(k) \quad k = 0,1,\ldots,N-1$$

Its DFT representation is:

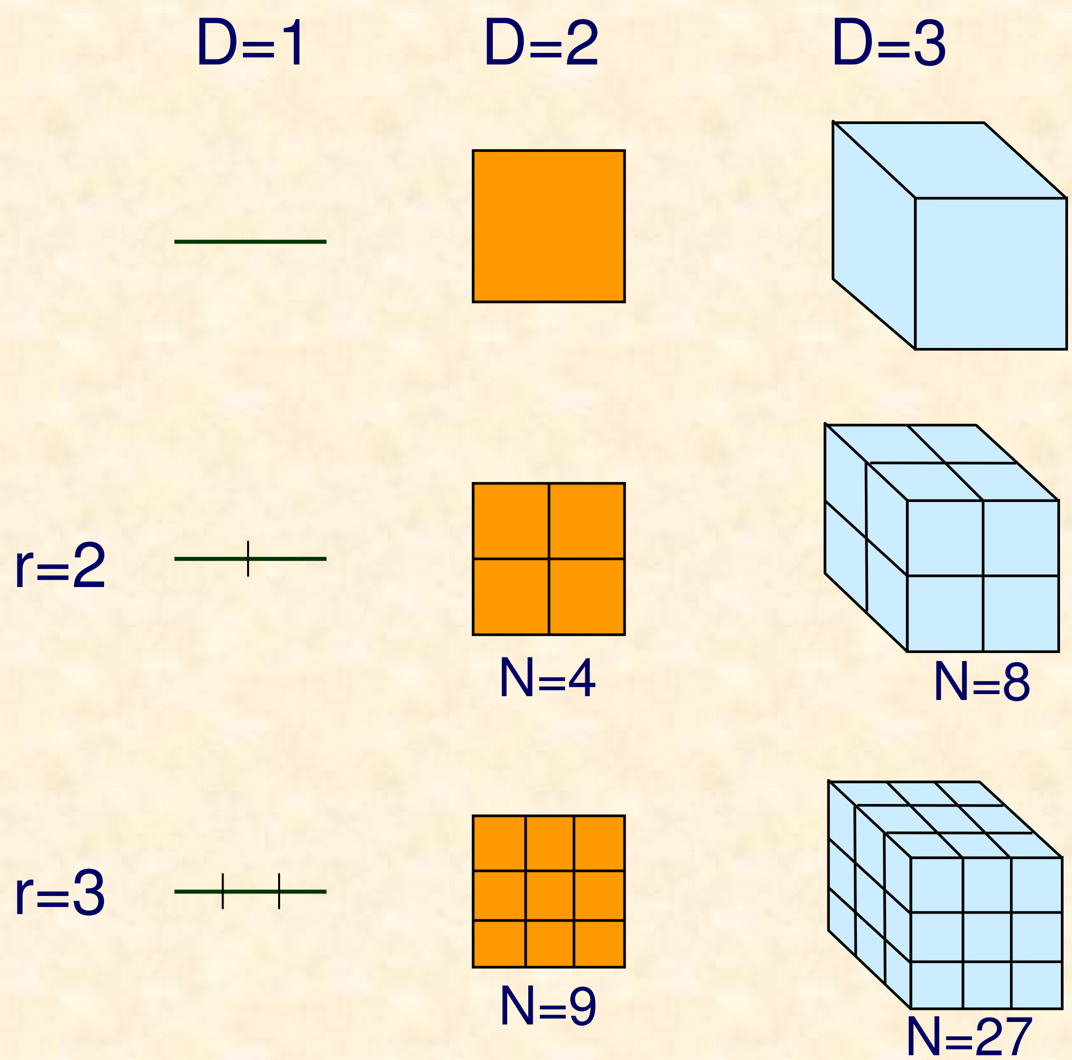
$$a(u) = \frac{1}{N}\sum_{k=0}^{N-1} s(k)\exp(-j2\pi uk/N), \quad u = 0,1,\ldots N-1$$

The complex coefficients *a(u)* are called the **Fourier descriptors** of the boundary.

Reconstruction of region boundary from first 5, 10 and 25 terms in Fourier expansion



Spatial dimensions – fractal dimensions



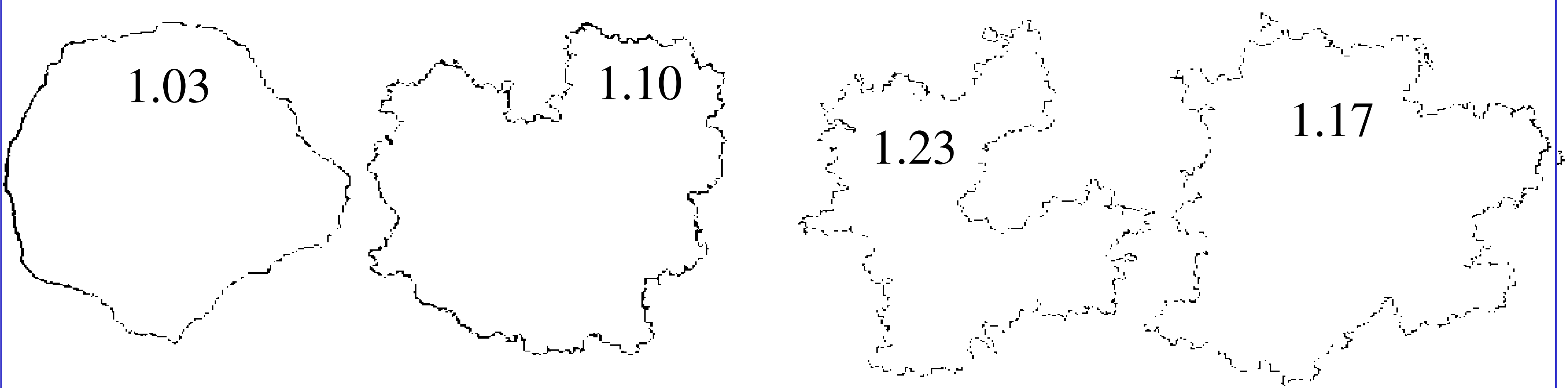
$$N = r^D$$

$$D = \frac{\log(N)}{\log(r)}$$

Does D need to be integer?

Fractal Dimension

The fractal dimension is the rate at which the length of an edge increases as the measurement scale is reduced. It may be understood as a measure of "roughness" of the edge or boundary.

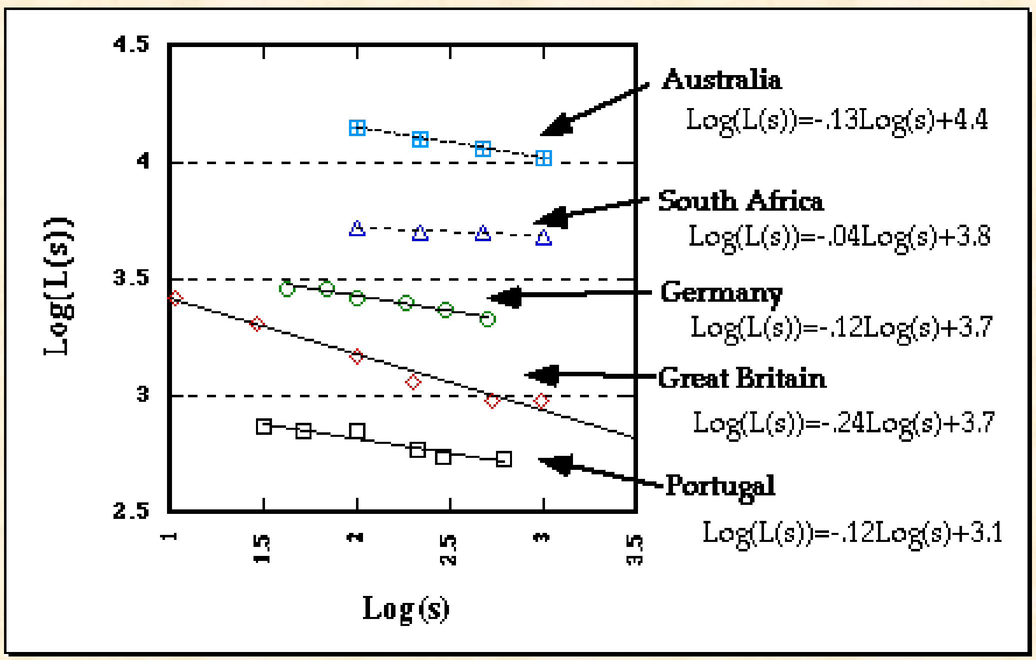


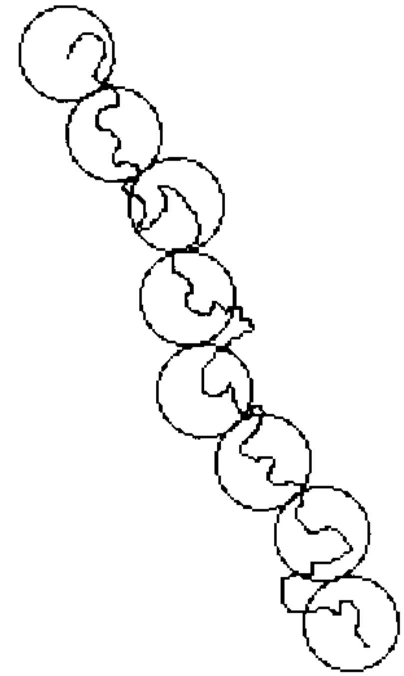
Several boundaries with different fractal dimension

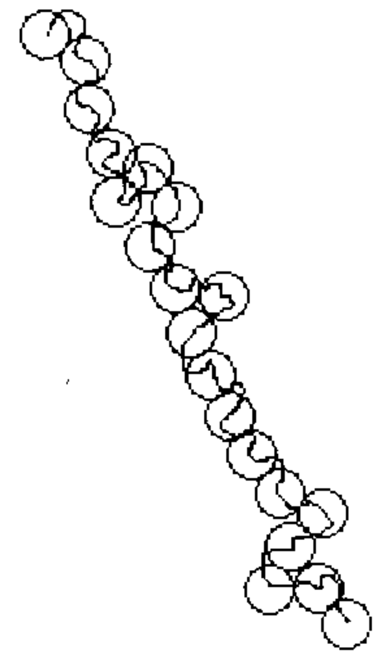
Fractal Dimension

<u>Richardson dimension</u> -

is obtained by counting the number of strides needed to walk along the boundary, as a function of stride length; the result plotted on log-log axes, is a straight line whose slope gives the fractal dimension.



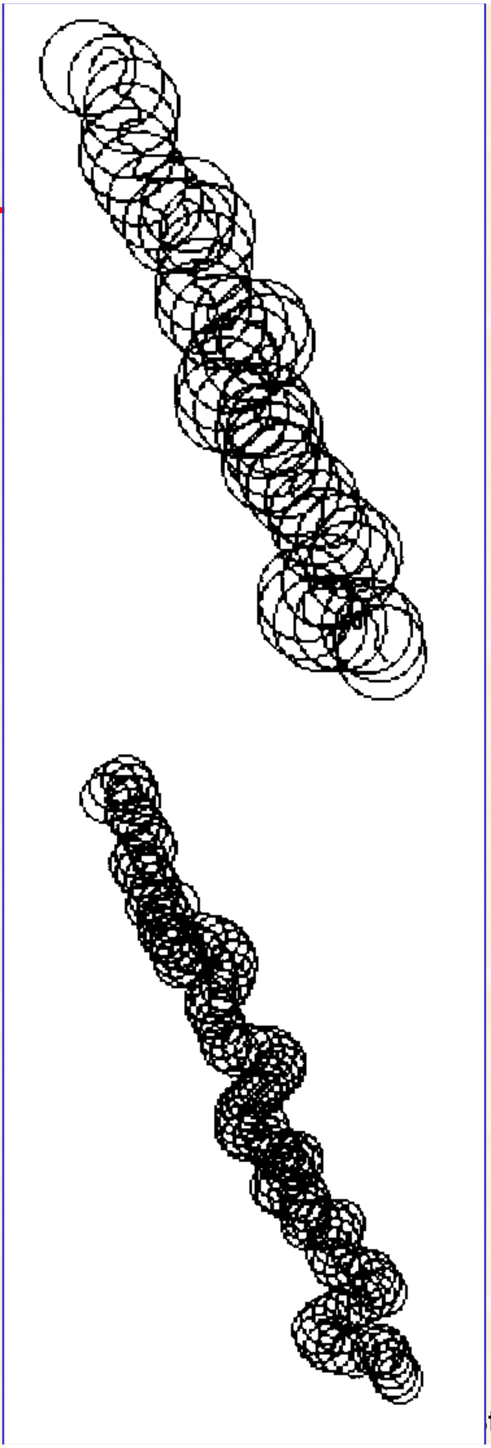




Fractal Dimension

Minkowski dimension -

in this method the circle is swept along a boundary (contour); plotting the area swept out by the circle (called sometimes the sausage) versus its radius on a log-log axes produces a line whose slope gives this fractal dimension.

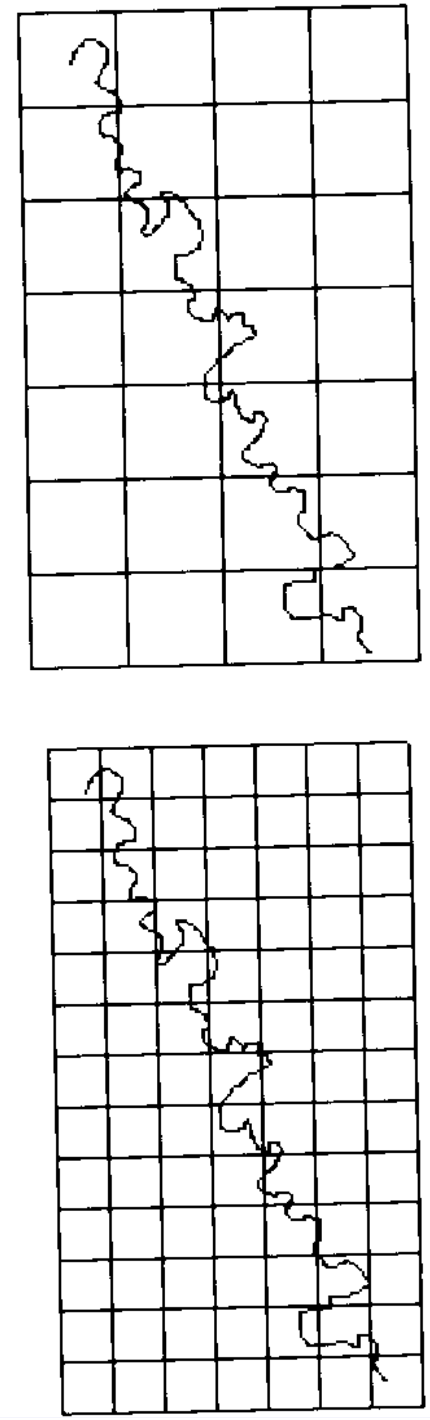


Fractal Dimension

<u>**Kolmogorov dimension**</u> -

a mesh of lines is drawn on the image and the number of grid squares through which the boundary passes is counted;

by plotting this number on log-log axes vs. the size of the grid and finding the slope of the line gives this fractal dimension.

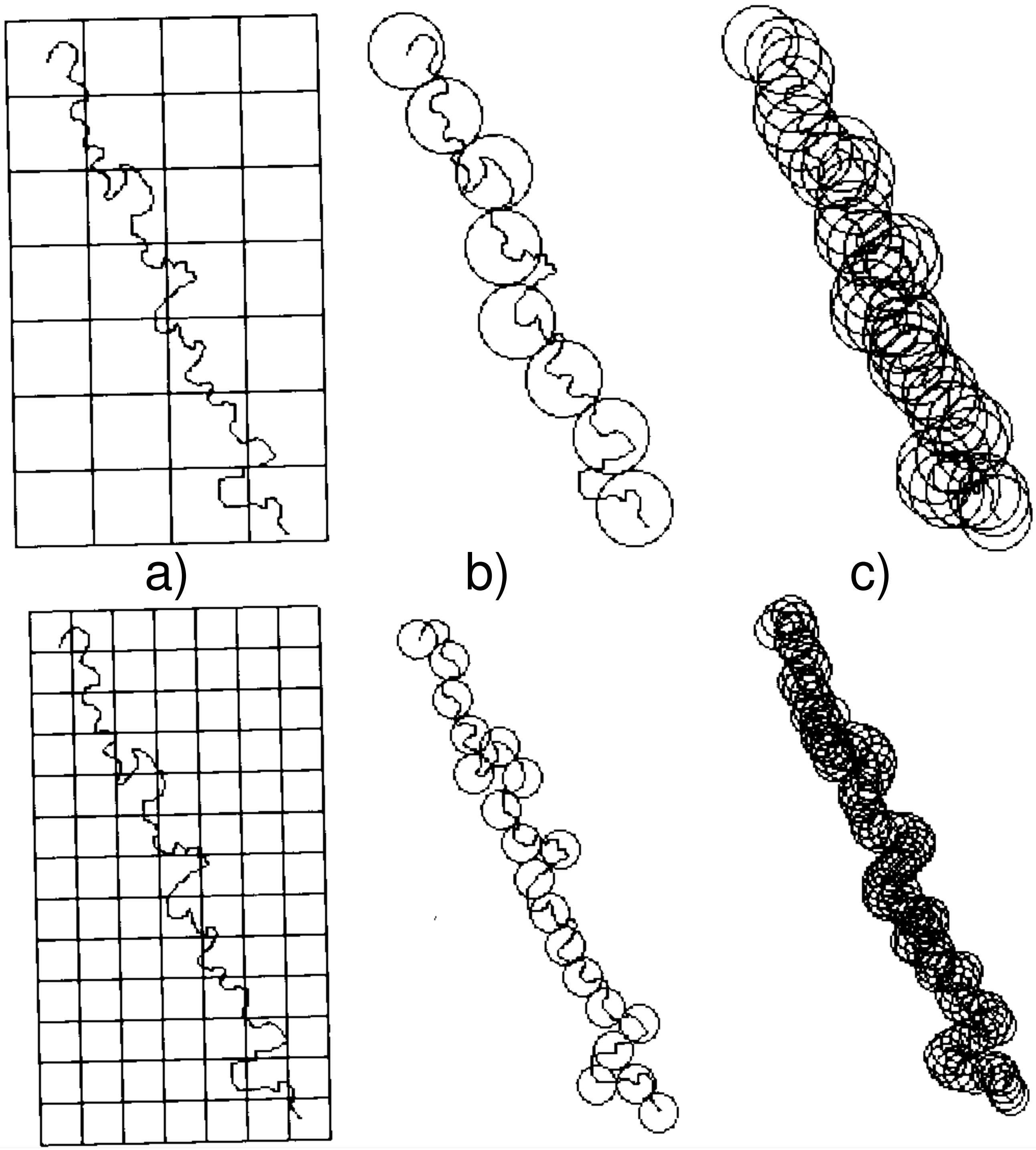


Fractal Dimension

a) Kolmogorov,

b) Richardson,

c) Minkowski.

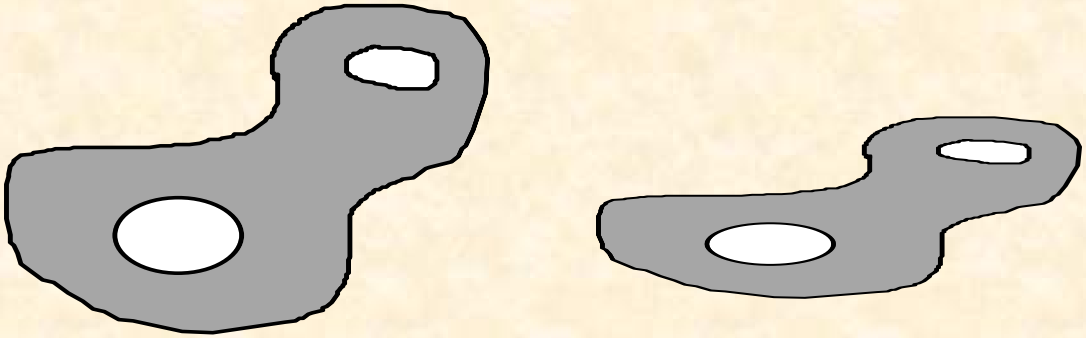


Topological shape features

Topologic shape features are invariant for „*rubber-sheet*” transformations of an elastic surface,

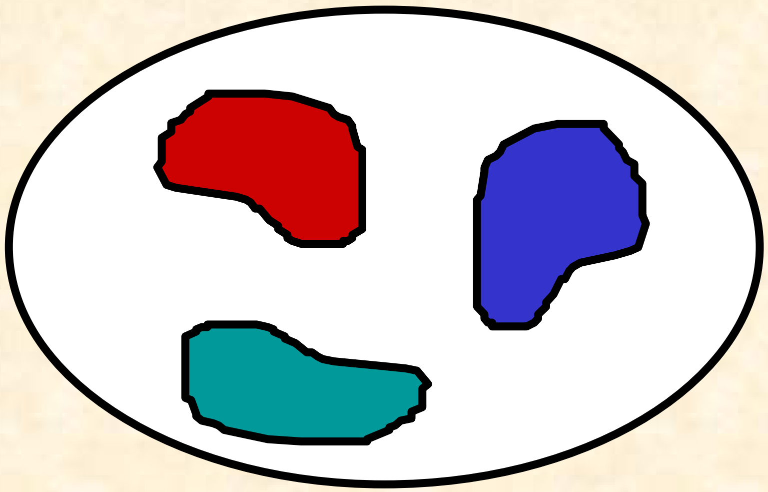
e.g., Euclidean distance, orthogonality do not belong to topological features

Topologic shape features offer a different approach to shape description used in geometrical features



Topological shape features

Connectivity of a region is a topological feature

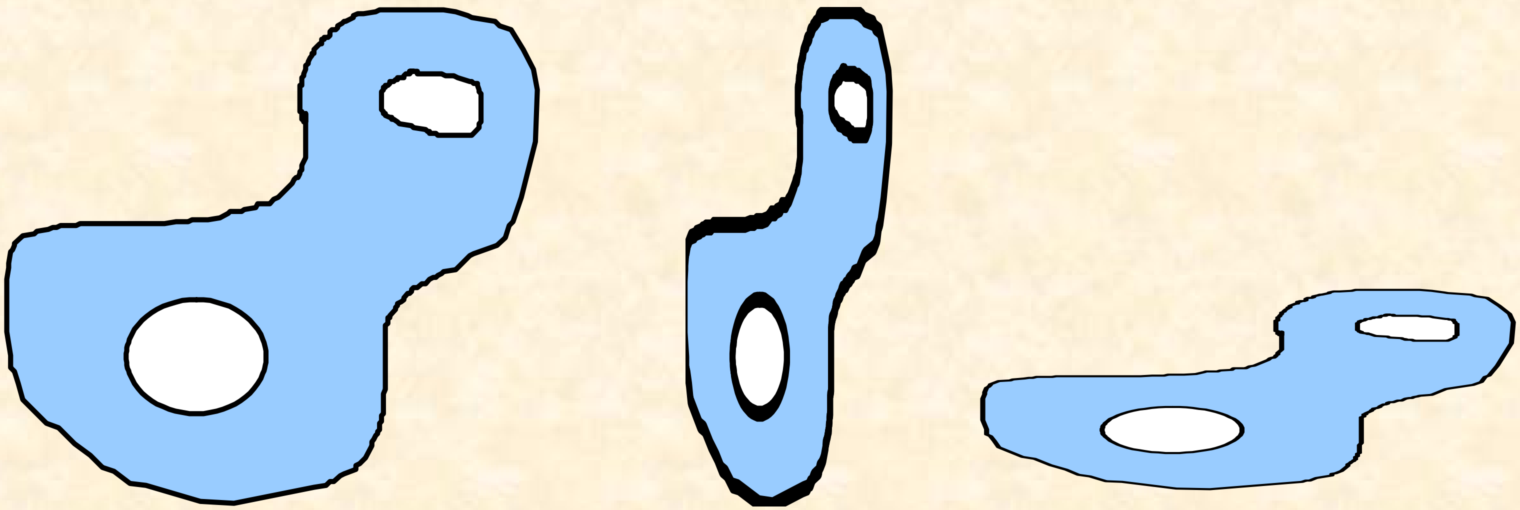


Region containing n=3 connected regions

M. Strzelecki, P. Strumiłło

Topological shape features

Number of holes in an object is a topological feature

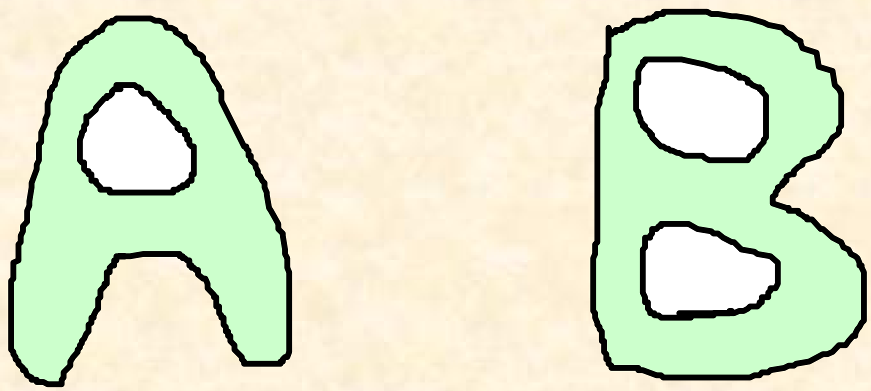


Objects with H=2 holes

Euler number

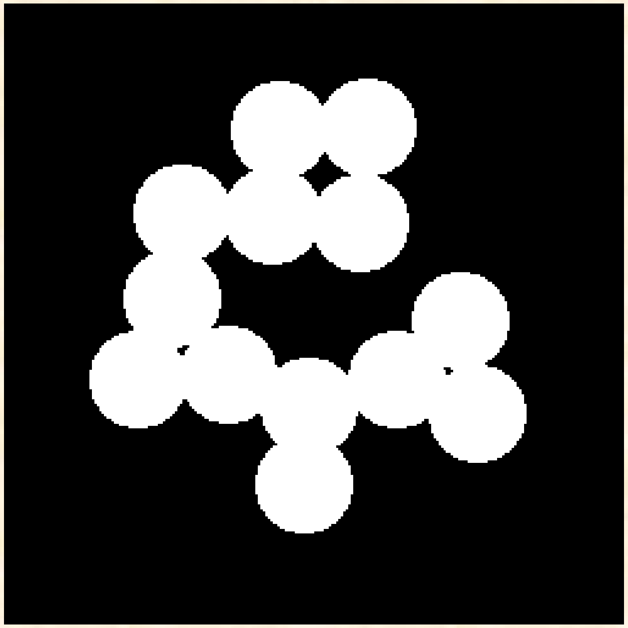
Euler number defines the relation between objects connectivity and number of holes $E = C - H$.

Euler number is a topological feature

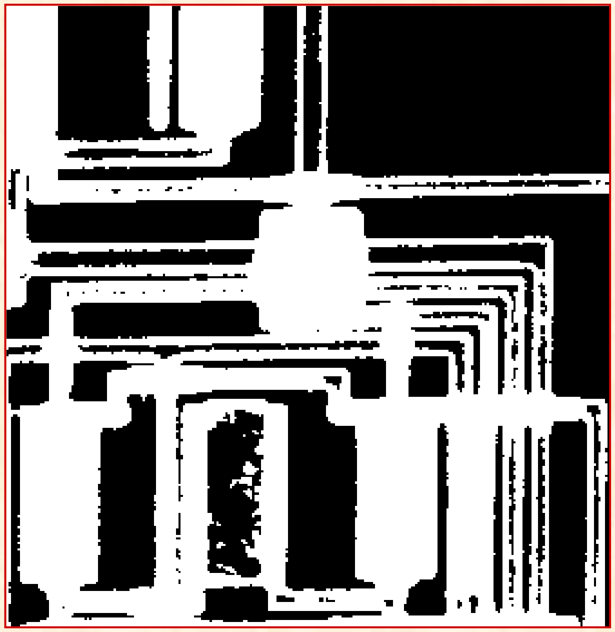


E=0 *E=-1*

Euler number - example



E=-2



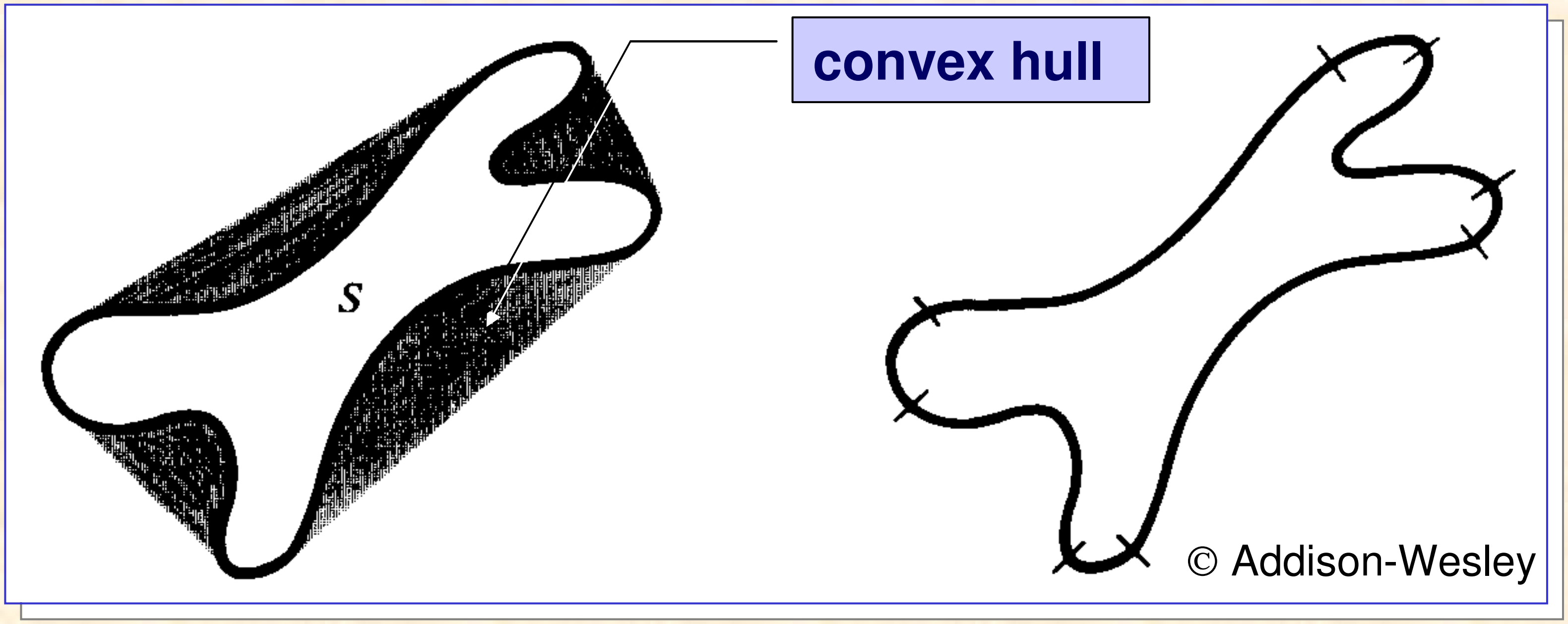
E=-85

```
%MATLAB
BW = imread('circles.tif');
imshow(BW);
E=bweuler(BW)
```

```
%MATLAB
BW1 = imread('circbw.tif');
imshow(BW1);
E=bweuler(BW1)
```

Convex hull

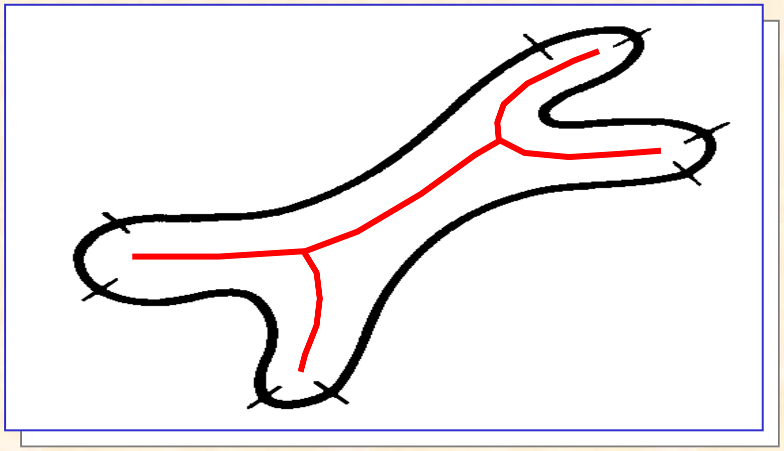
Convex hull C_H of object S is the smallest area region for which the union $C_H \cup S$ is a convex shape.



Skeleton

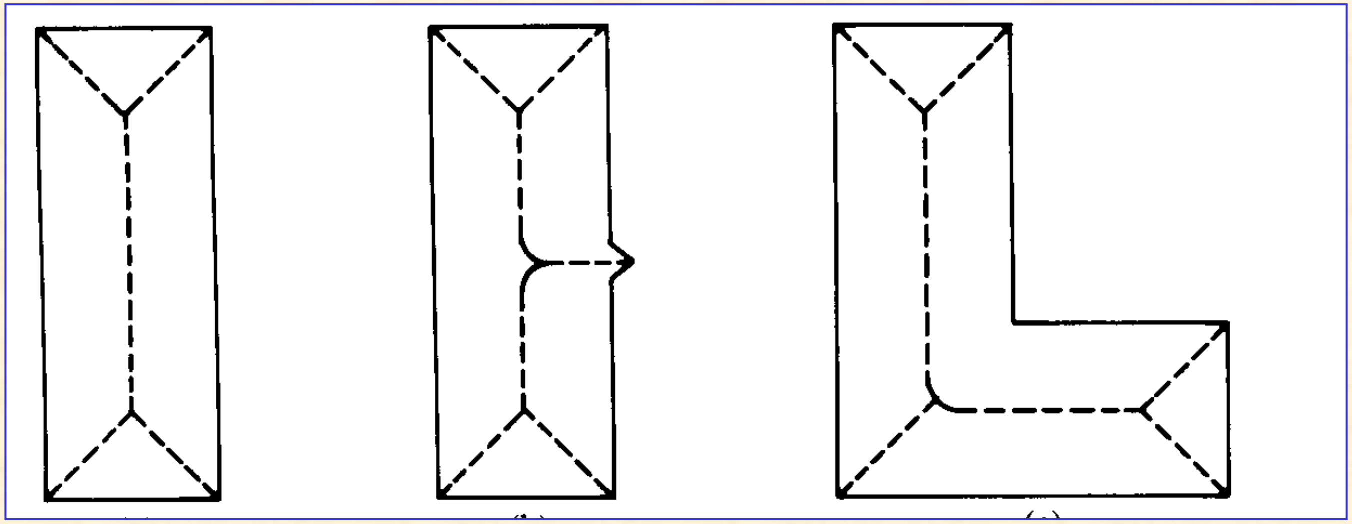
An important way of shape description is reducing the size of the object to its „skeleton”.

The concept of object skeleton is frequently used in image analysis applications, e.g. optical character recognition, edge tracking etc.



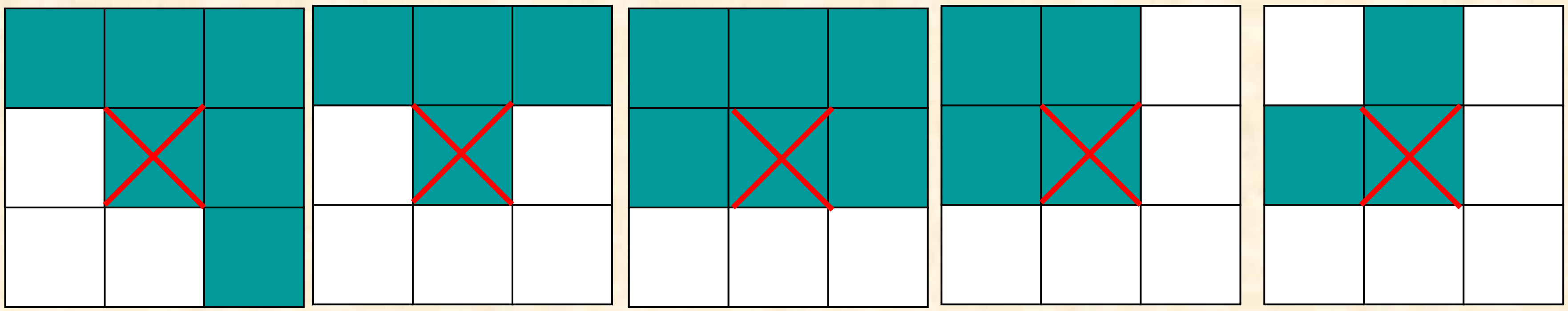
Skeletonization

Skeleton of an object can be obtained be means of ***medial axis transformation***.

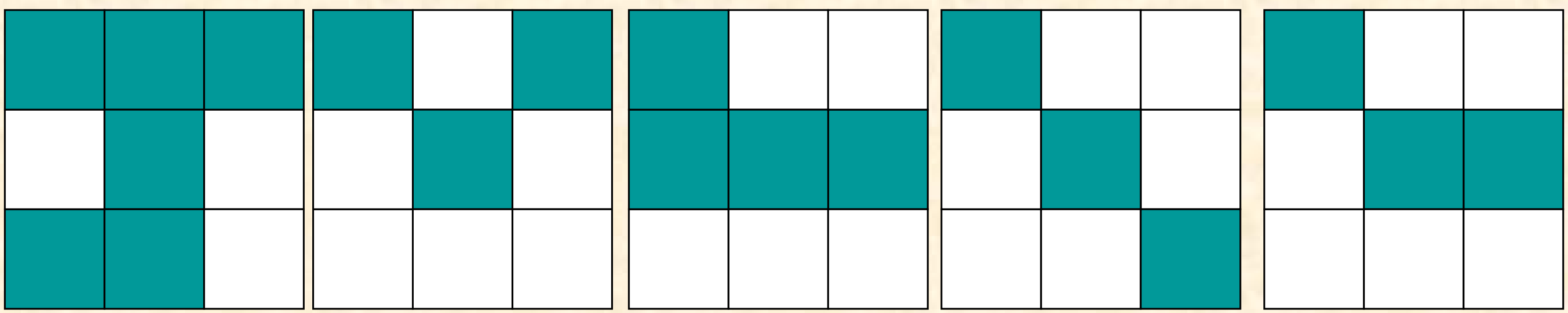


Skeletons of example shapes

Skeletonization

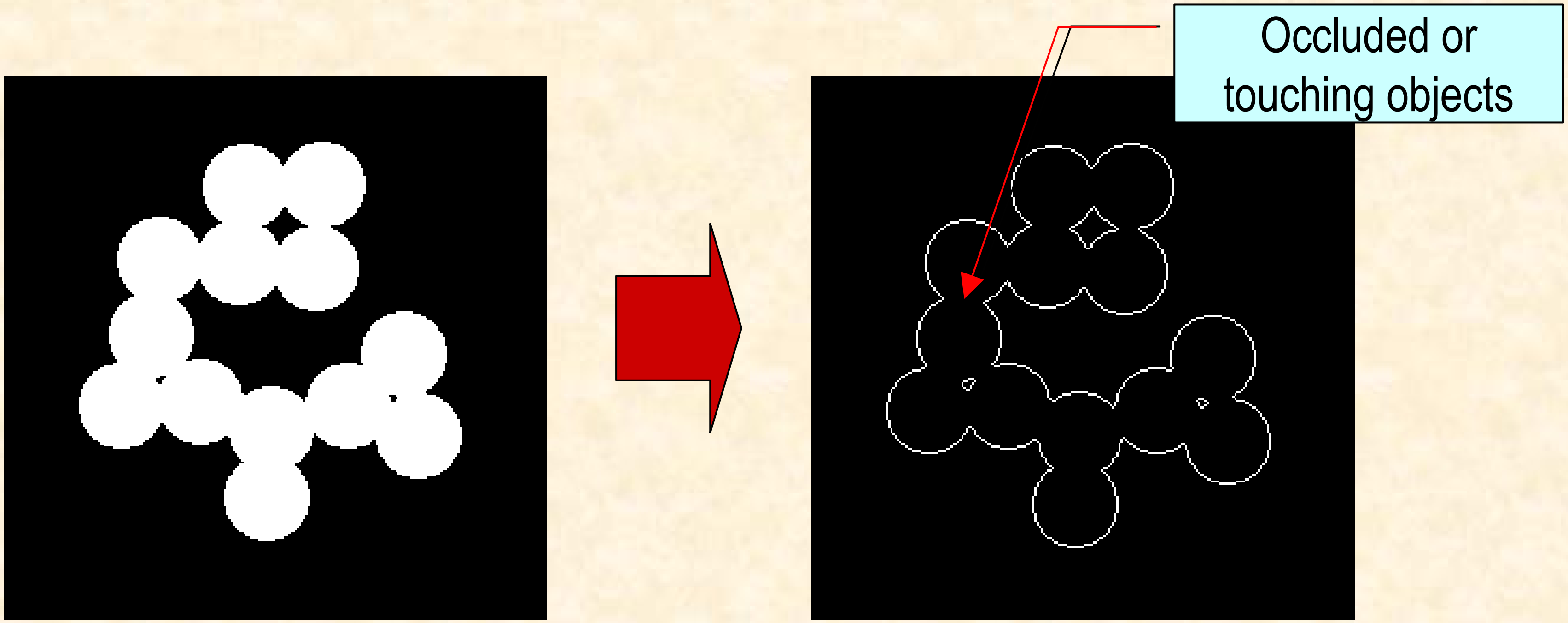


Patterns for which the central pixel **can** be removed



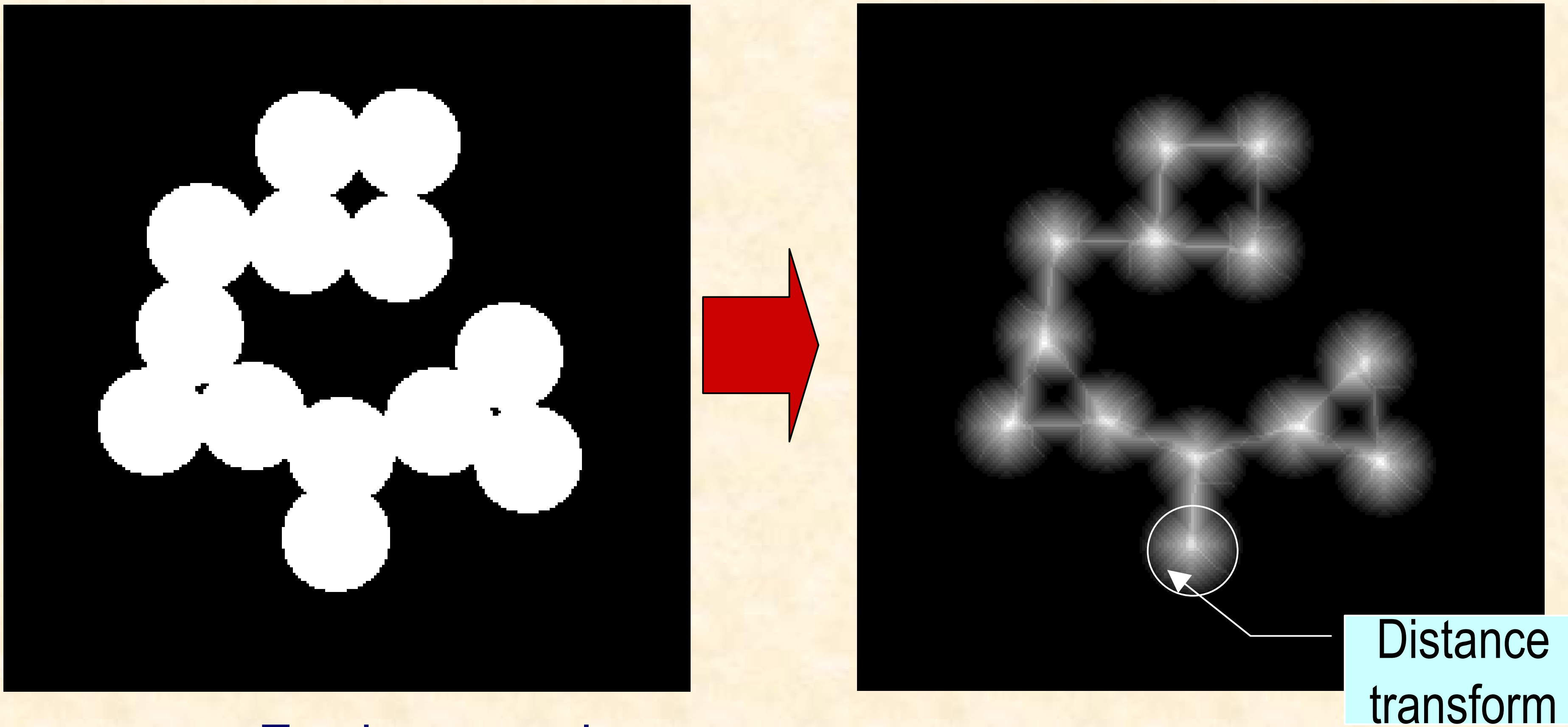
Patterns for which the central pixel **cannot** be removed

Object counting



Edge detection does not solve the problem

Object counting

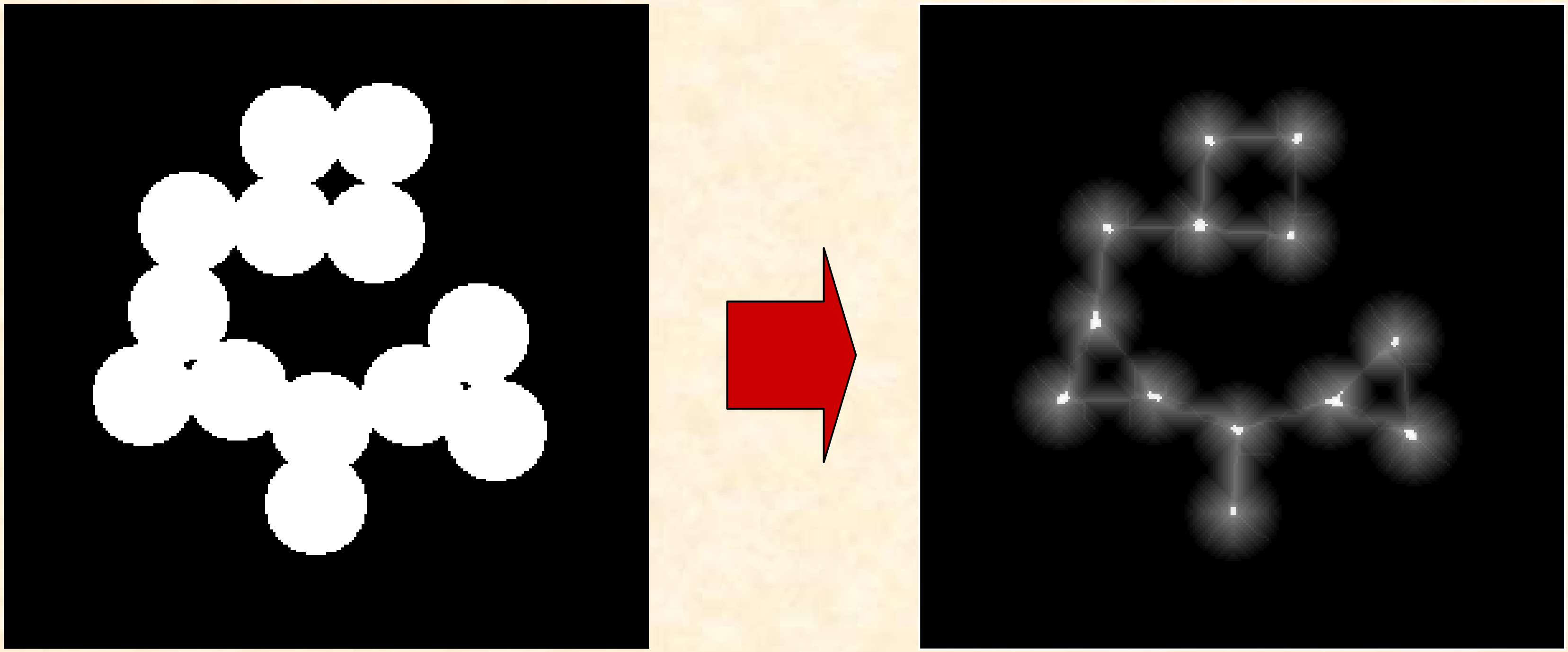


Erode many times:

BW(i+1)=bwmorph(BW(i),'erode',1);

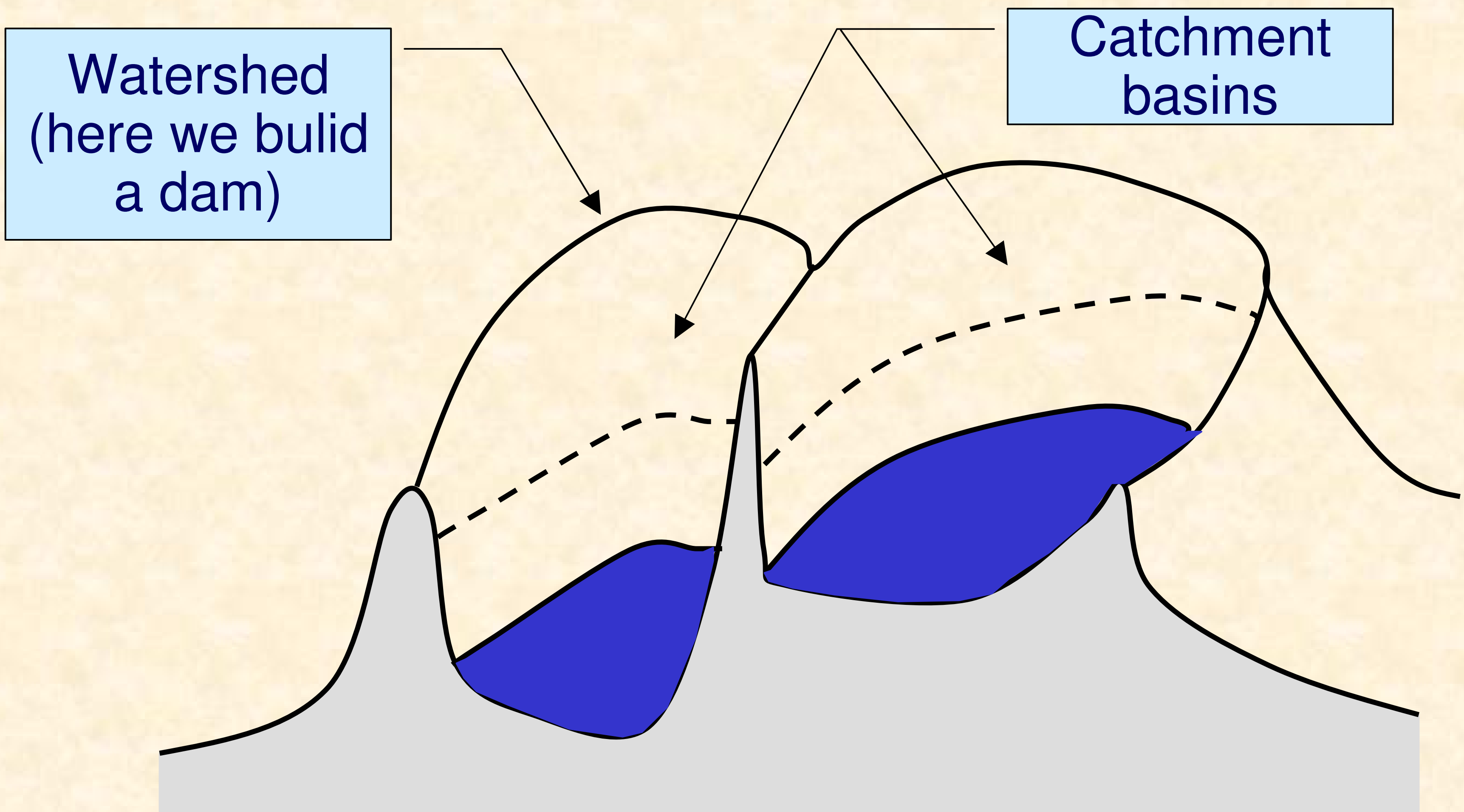
and add all intermediate images

Object counting

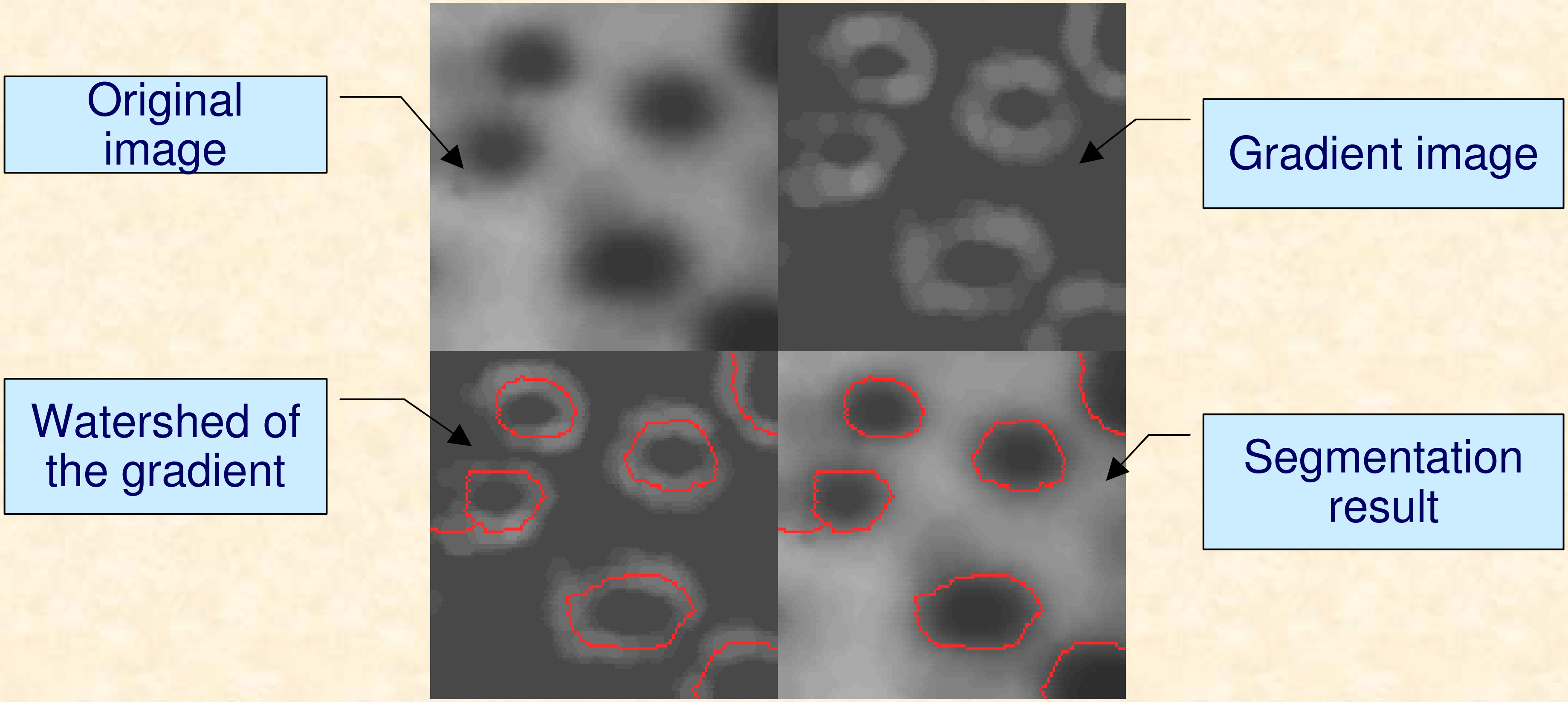


Threshold the multiple eroded image

Watershed segmentation



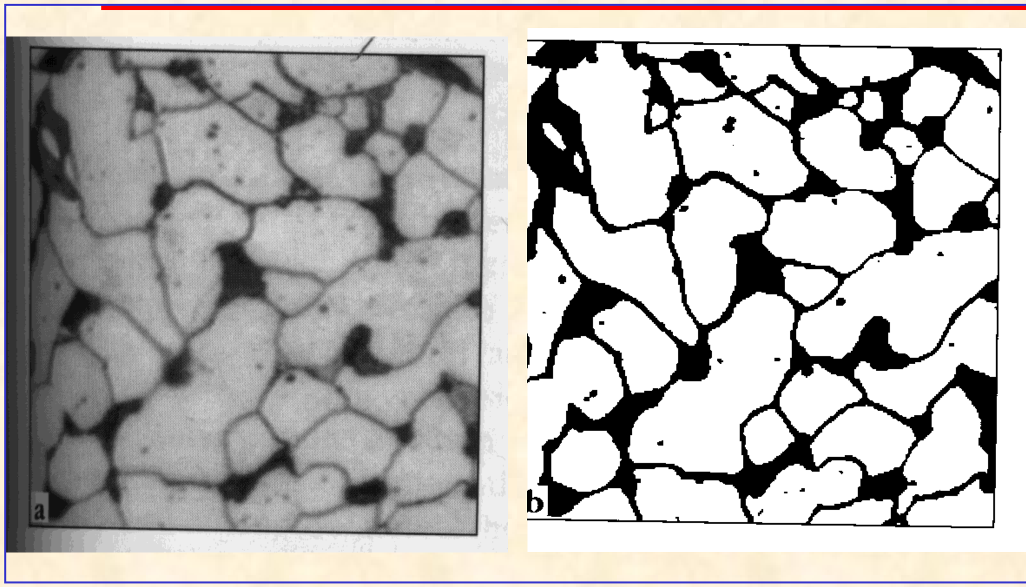
Watershed segmentation



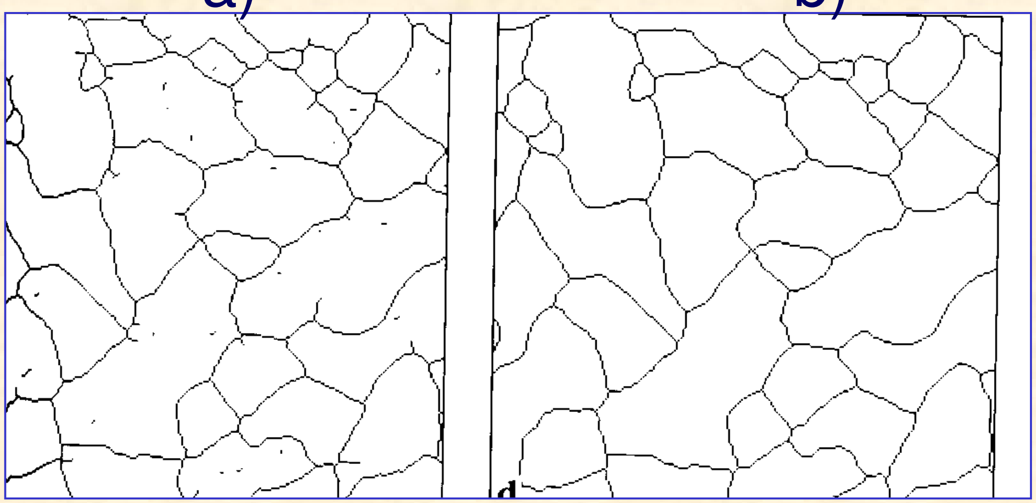
S. Beucher, "The watershed transormation applied to image segmentation", *Conference on Signal and Image Processing in Microscopy and Microanalysis*, pp. 299-314, September 1991.

website: http://cmm.ensmp.fr/~beucher/publi/pfefferkorn.pdf

Object skeleton - applications



a) b)



c) d)

Microscopic images of steel (with carbon particles):

a) source image

b) thresholded image

c) after skeletonizing

d) after pruning.