Lecture 6
Object representation and description

Robin Strand

GW 11.1-11.4

Suggested problems: 11.19,11.25

Image analysis fundamental steps

image acquisition

preprocessing,
enhancement

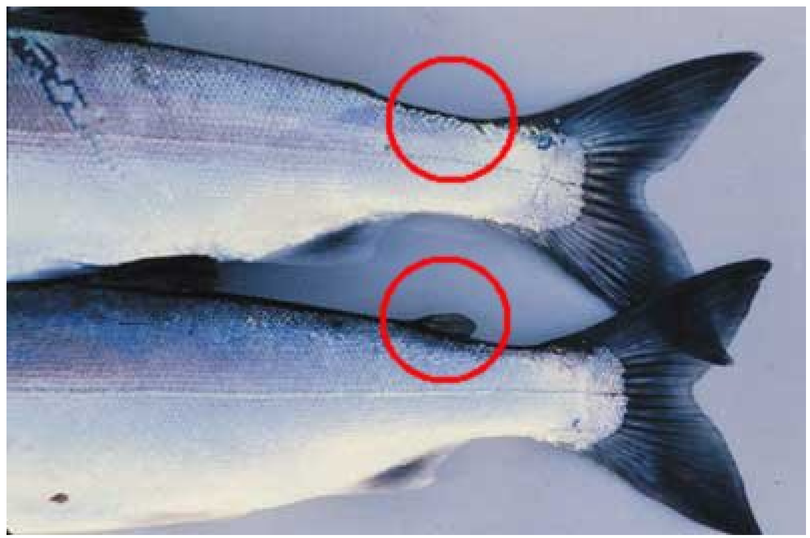
segmentation

Representation, description,
feature extraction

classification,
interpretation,
recognition

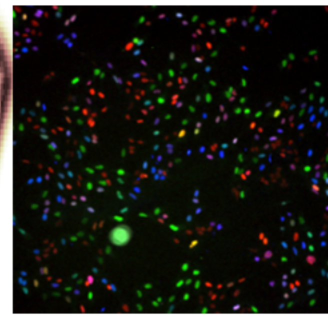
result

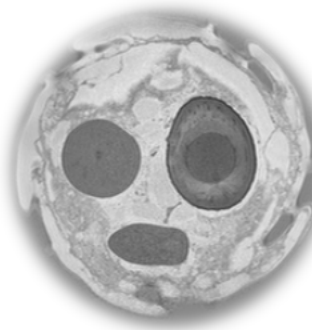
Farmed vs wild salmon

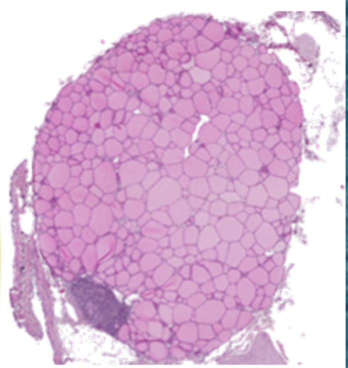


Distinguishing between salmon and sea trout

Representation and description







Representation and description

Commonly after segmentation one needs to **represent** objects in order to **describe** them

- External (boundary):
  - Representation: Polygon of the boundary
  - Description: The circumference
- Internal (regional)
  - Representation: Pixels inside the object
  - Description: The average color

Representations and descriptors

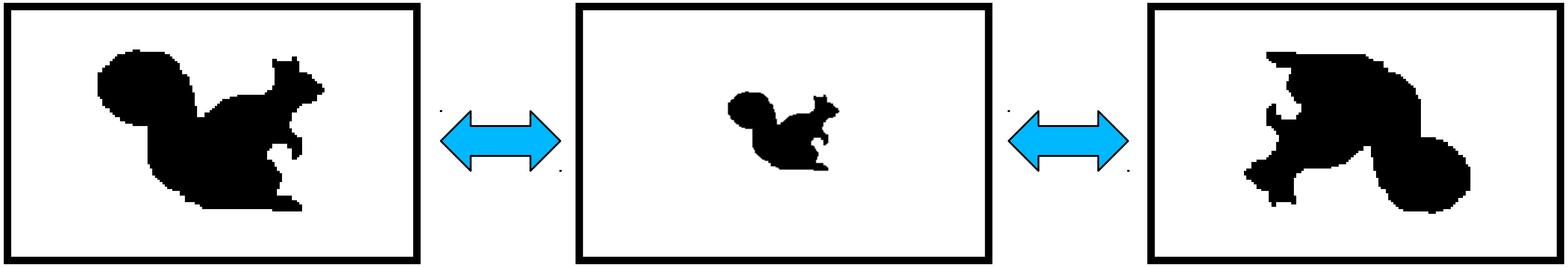
- The Representation of the Object
    - An encoding of the object
    - Truthful but possibly approximate
- A Descriptor of the Object:
    - Only an aspect of the object
    - Suitable for classification
    - Consider invariance to e.g. noise, translation,

Shape Representation

- Sometimes necessary/desirable to represent an object in a less complicated or more intuitive way
- Simple descriptions like enclosing circle, enclosing rectangle, inscribed circle etc.
- The boundary or boundary segments
- Divide an object into regions or parts
- Represent by ”skeleton”

Scale, rotation and translation

- Often we want descriptors that are invariant of scale, rotation and translation:

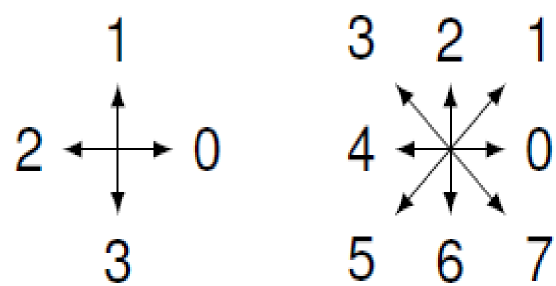


- However, not always. In Optical Character Recognition (OCR) rotation and scale is important (e.g. ‘P’ and ‘d’)

Chain code: a contour based shape representation

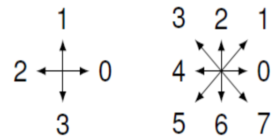
Chain code – the sequence of steps generated when walking around the boundary of a segmented region

Chain code can be defined for 4 and 8 neighbours

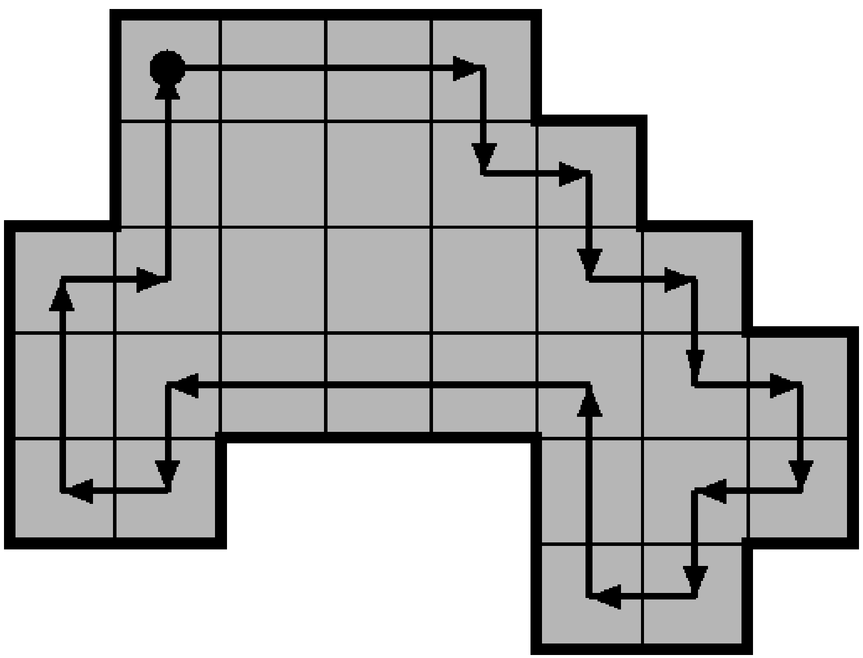


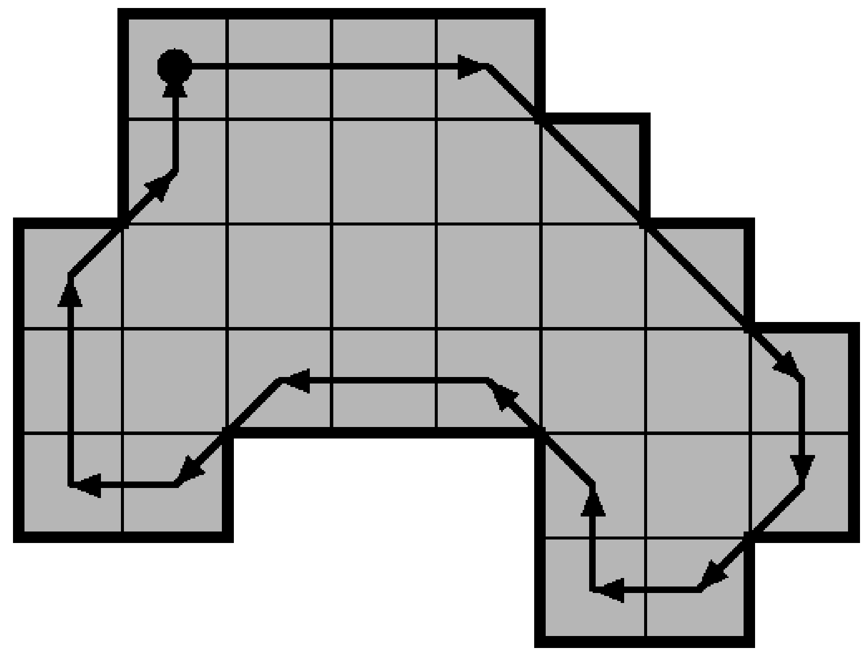
Chaincode example

4-connected:
0003030303232
11222232110111



8-connected:
0007776542344542212





Chain Coding issues/drawbacks

- Code becomes very long and noise sensitive

  ->Use larger grid spacing, smooth/edit the code

- Scale dependent

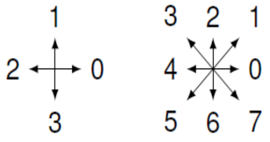
  ->Choose appropriate grid spacing

- Start point determines result

  ->Treat code as circular (minimum magnitude integer)
  754310 -> 075431

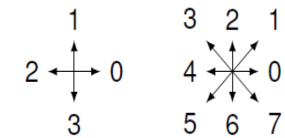
- Depends on rotation

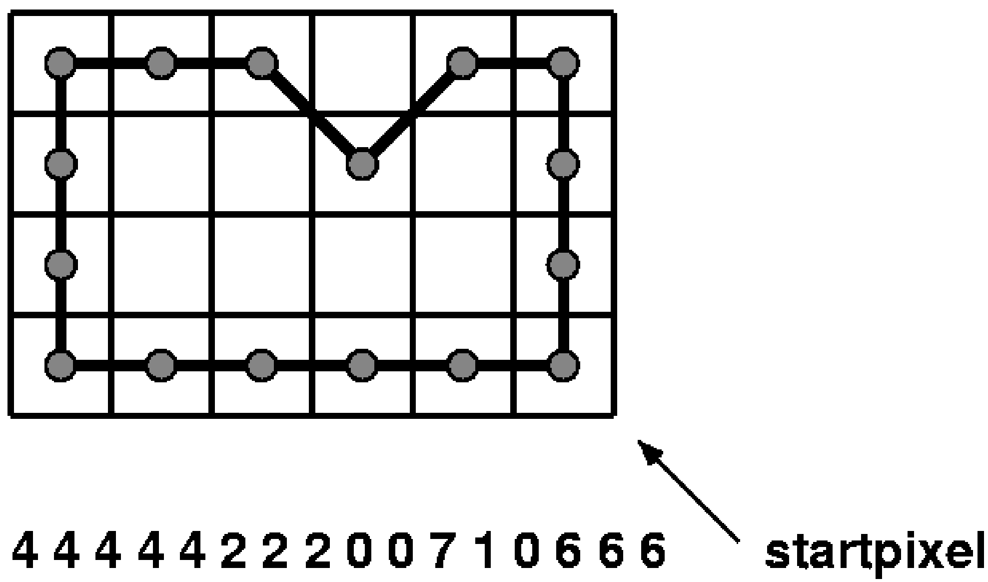
  ->Calculate difference code (counterclockwise)
  075431 ->



Example: editing the chain code

replace 0710 with 0000



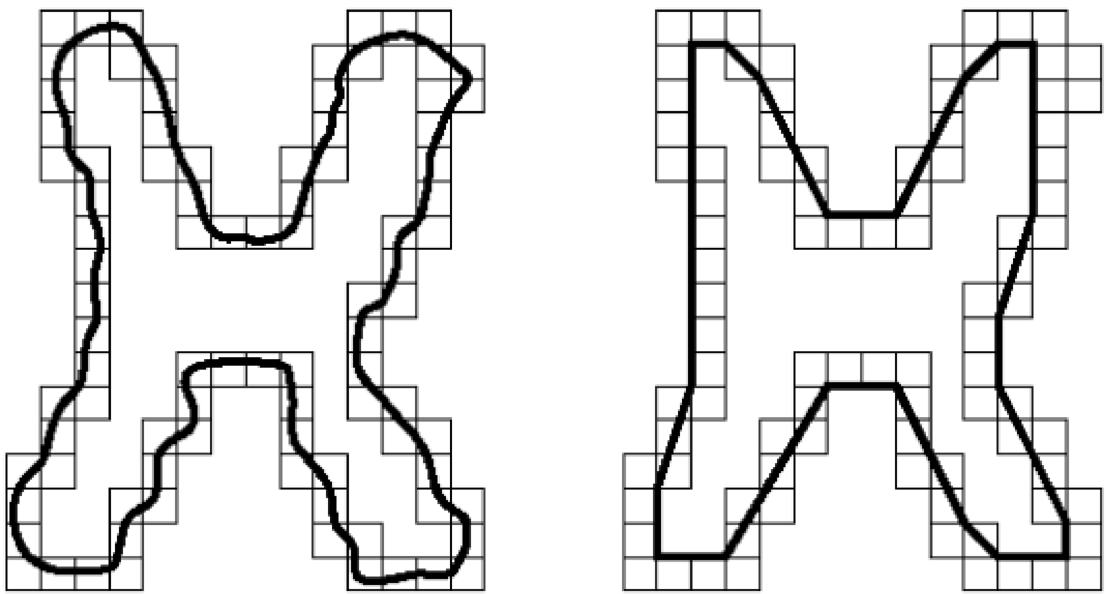


Polygonal Approximations

- A digital boundary can be approximated (simplified)
- For closed boundaries:
  - Approximation becomes exact when no. of segments of the polygons is equal to the no. of points in the boundary
- Goal is to capture the essence of the object shape
- Approximation can become a time consuming iterative process

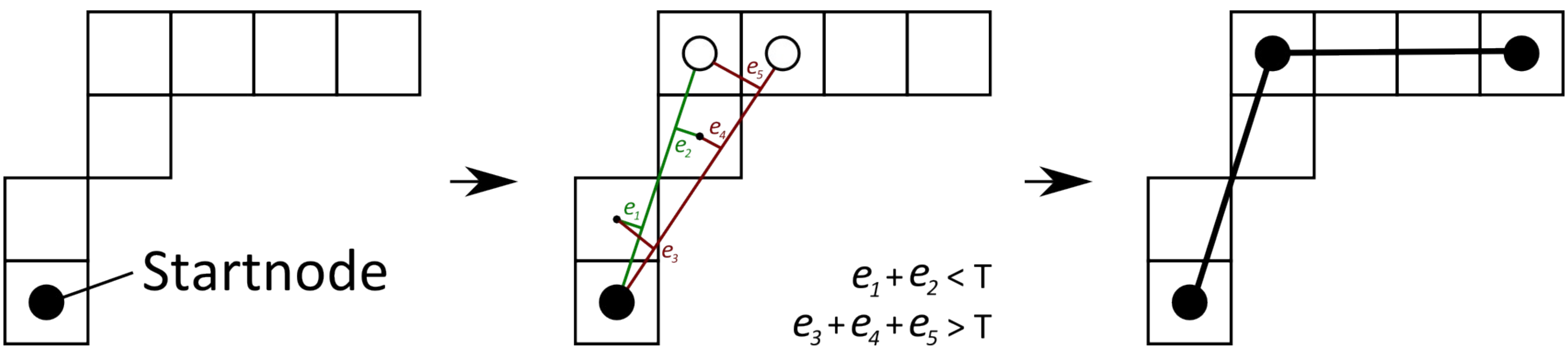
Polygonal Approximations

- **Minimum Perimeter Polygons** (MPPs)
  - Cover the boundary with cells of a chosen size and force a rubber band like structure to fit inside the cells



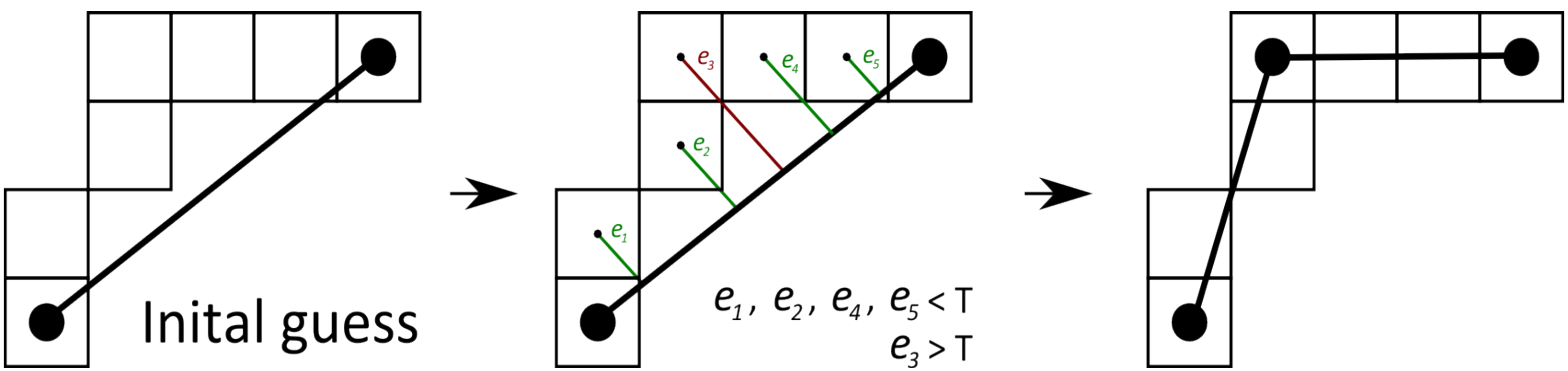
Polygonal Approximations

- **Merging techniques**
  1. Walk around the boundary and fit a least-square-error line to the points until an error threshold is exceeded
  2. Start a new line, go to 1
  3. When the start point is reached the intersections of adjacent lines are the vertices of the polygon

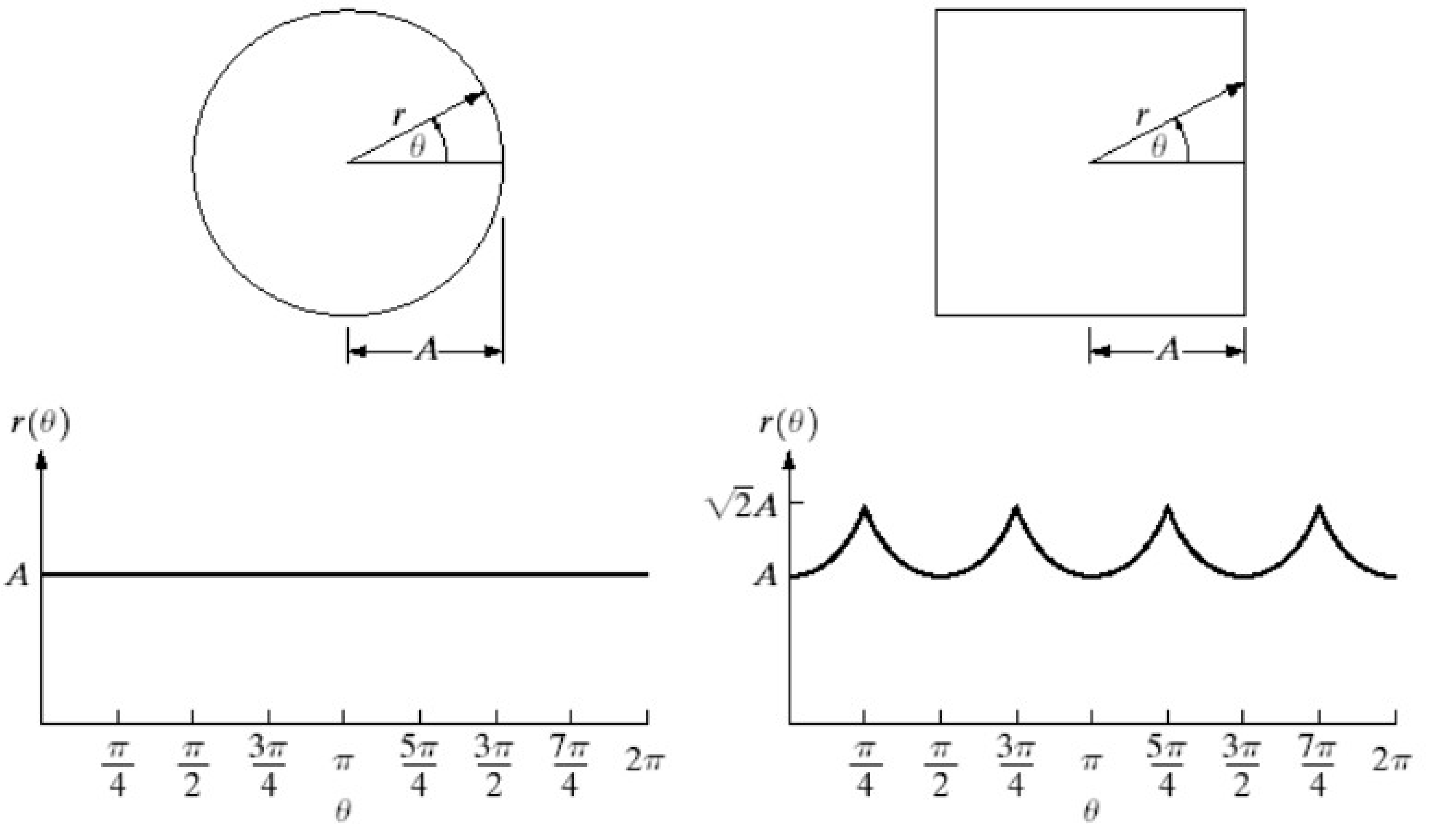


Polygonal Approximations

- **Splitting techniques**
  1. Start with an initial guess
  2. Calculate the orthogonal distance from lines to all points
  3. If maximum distance > threshold, create new vertex there
  4. Repeat until no points exceed criterion



Boundary representation: signatures



Signatures

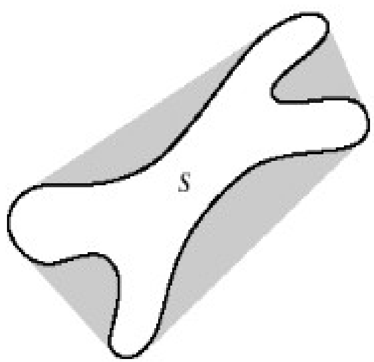
- A 1D representation of a boundary
- Could be implemented in different ways
  - Distance from centre point to border as a function of angle
  - Angle between the tangent in each point and a reference line (histogram of this is called slope density function)
- Independent of translation, but not rotation & scaling.

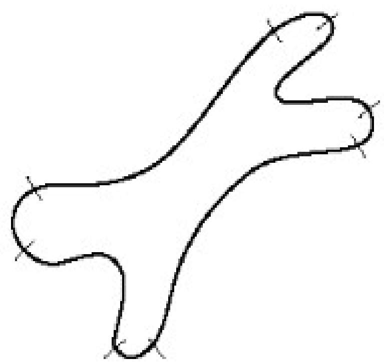
->Select unique starting point (e.g. based on major axis)

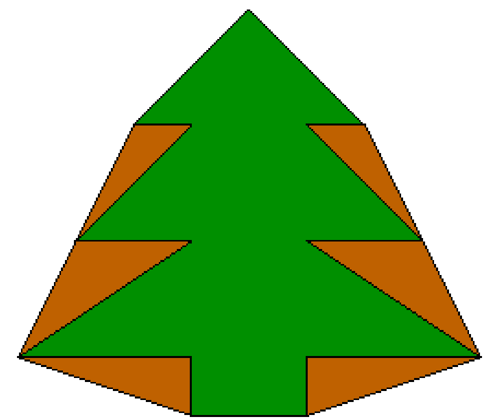
->Normalize amplitude of signature (divide by variance)

Boundary segments

- **When a boundary contains major concavities that carry shape information it can be worthwhile to decompose it into segments**
- **A good way to achieve this is to calculate the convex Hull of the region enclosed by the boundary = minimal enclosing convex region**







->Smooth prior to Convex hull calculation

->Calculate Convex Hull on polygon approximation

Convex hull, deficiency and concavity tree

Convex Hull = minimal enclosing convex region

Convex region = all points can be connected through a straight line inside the region

Convex deficiency = Convex hull – object

The number and distribution of convex deficiency regions may also be useful

=› Concavity tree, generate convex hulls and deficiencies recursively to create at concavity tree

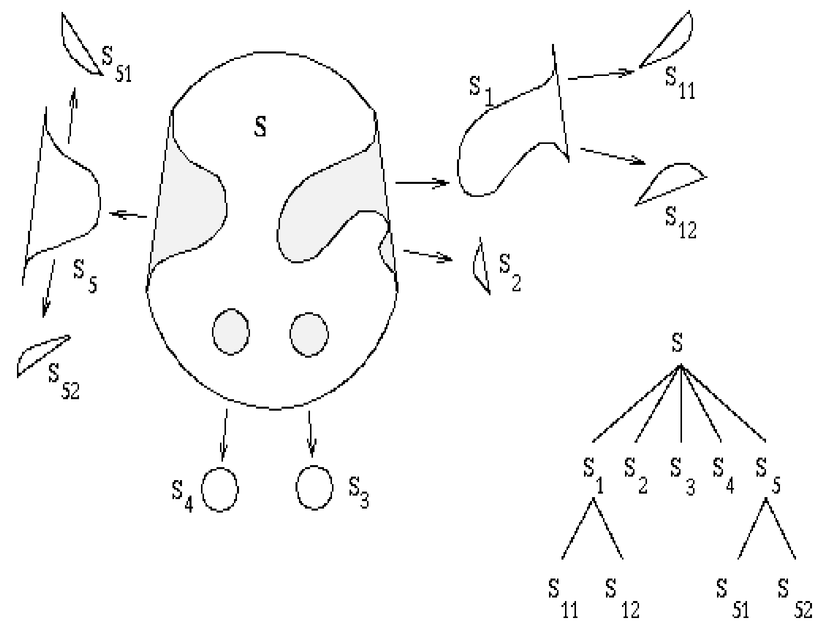


Figure 6.30 *Concavity tree construction: (a) Convex hull and concave residua, (b) concavity tree.*

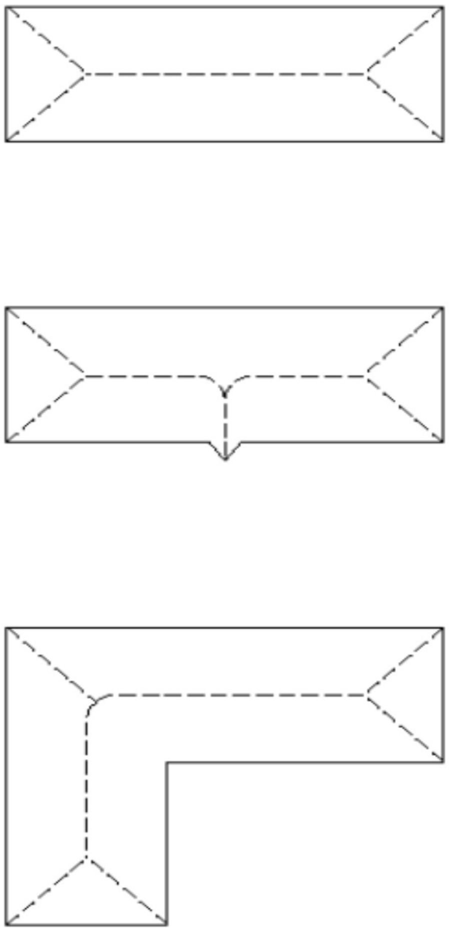
Skeletons

“Curve representation” of the object

Should in general be thin, centered, topologically equivalent to original object and reversible

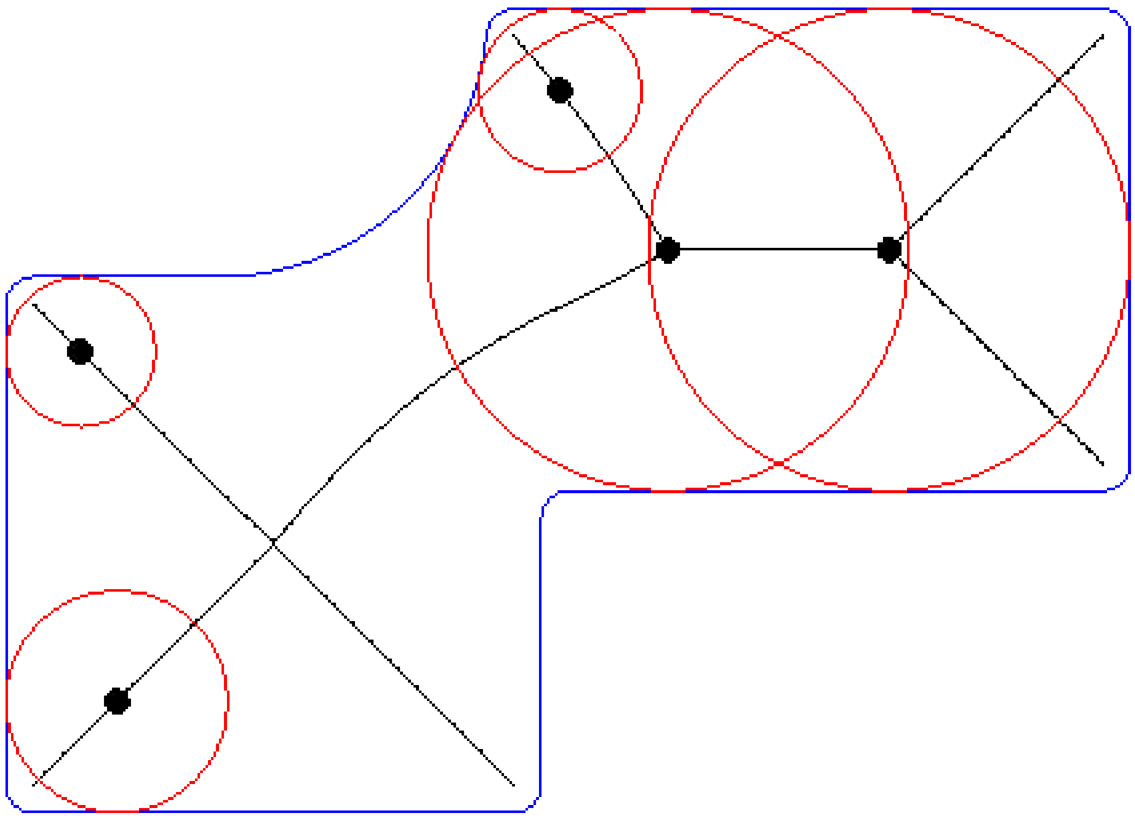
Can be created by thinning =iteratively removing pixels from the border while keeping the overall shape and topology (see book for detailed description)
or by medial axis transform (MAT) = all inscribed circles touching two or more points at the border at the same time

Skeletons are sensitive to small changes in shape

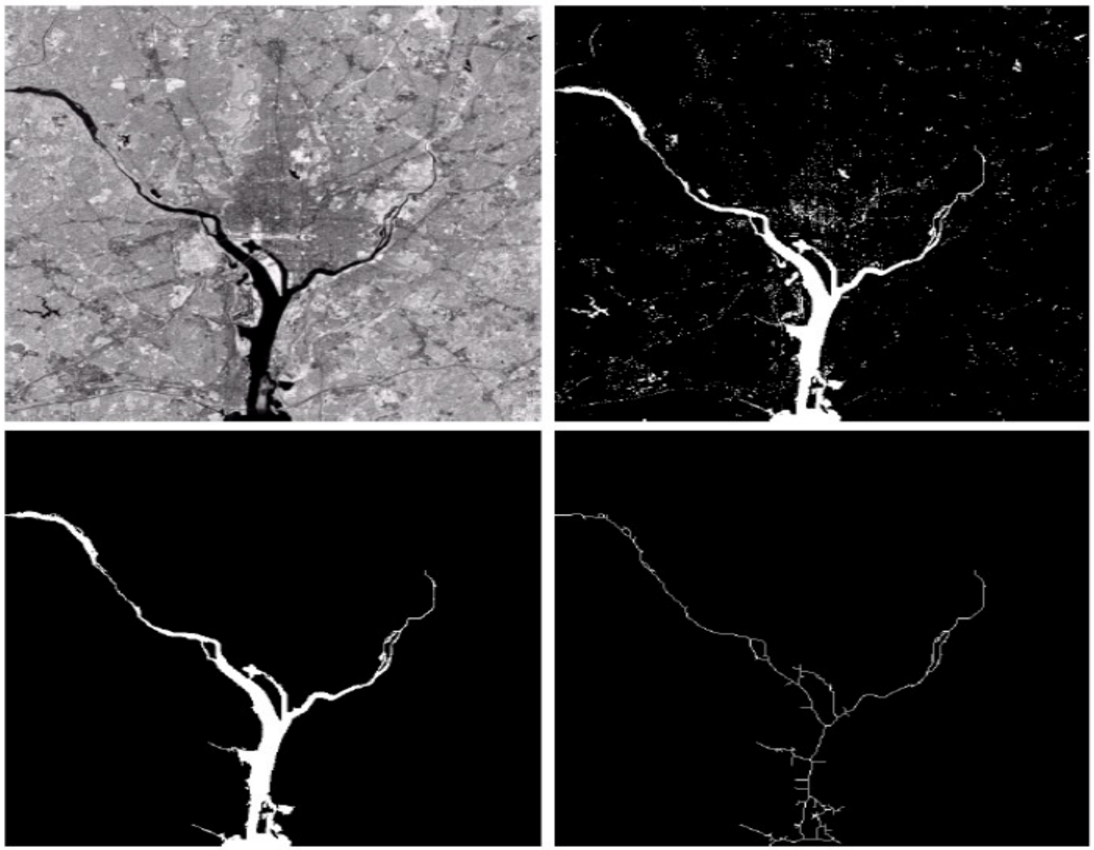


- >smooth first or ”prune” skeleton afterwards

Skeleton from medial axis



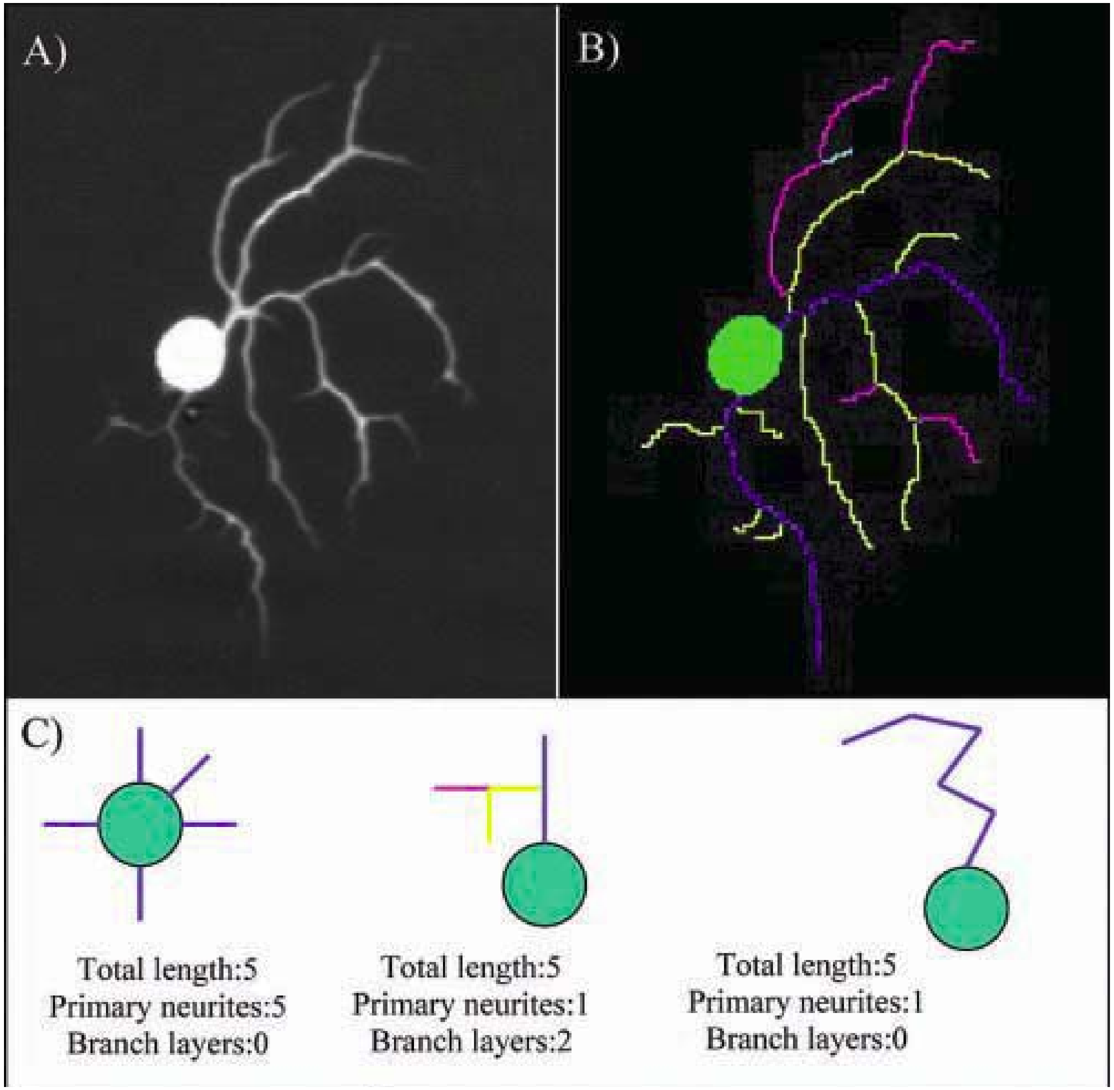
Skeleton example



Largest connected component is chosen as object of interest

Skeleton or medial axis representation used for length measurements

Skeleton example: Neurite outgrowth analysis



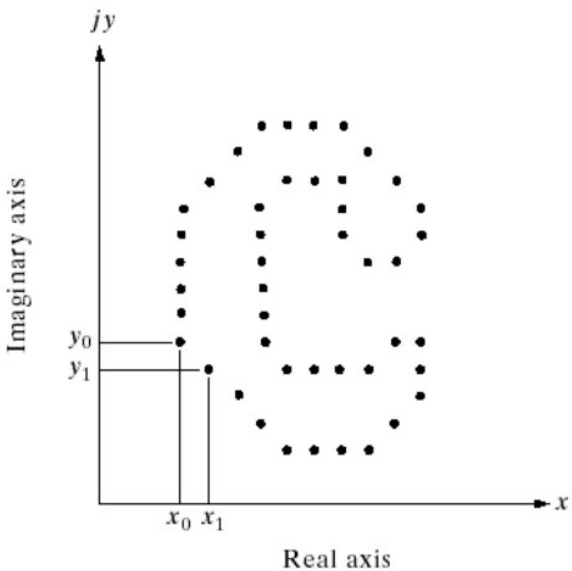
Descriptors

- After representation, the next step is to **describe** our boundaries and regions so that we later can **classify** them (next lecture)
- A description is an aspect of the representation
- What descriptor is useful for classification of
    - adults / children
    - pears / bananas / tomatoes

Simple boundary (segment) descriptors

- Length (perimeter)
- Diameter = $\max_{i,j}\left[D(p_i, p_j)\right]$ = major axis
- Minor axis (perpendicular to major axis)
- Basic rectangle = major × minor
- Eccentricity = major / minor
- Curvature= rate of change of slope

Fourier descriptors



- Represent the boundary as a sequence of coordinates
- Treat each coordinate pair as a complex number

$$s(k) = [x(k), y(k)]\,, k = 0, 1, 2, \ldots, K-1$$

$$s(k) = x(k) + iy(k)$$

Fourier descriptors

- From the DFT of the complex number we get the Fourier descriptors (the complex coefficients, a(u))

$$a(u) = \sum_{k=0}^{K-1} s(k)e^{-j2\pi uk/K}, u = 0, 1, 2, \dots, K-1$$

- The IDFT from these coefficients restores s(k)

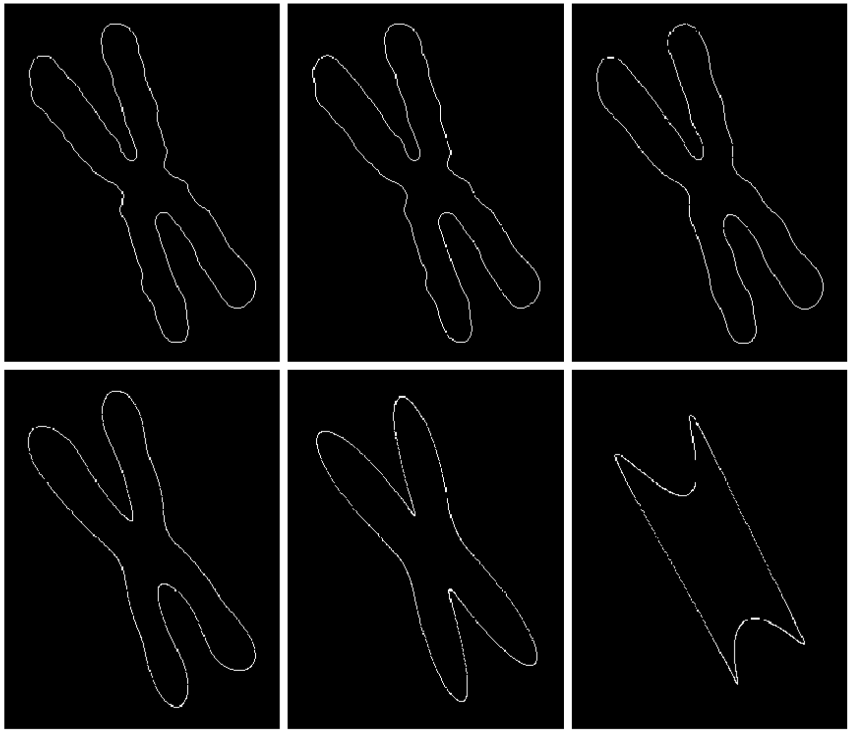
$$s(k) = \frac{1}{K}\sum_{u=0}^{K-1} a(u)e^{j2\pi uk/K}, k = 0, 1, 2, \dots, K-1$$

- We can create an approximate reconstruction of s(k) if we use only the first P Fourier coefficients

$$\hat{s}(k) = \frac{1}{P}\sum_{u=0}^{P-1} a(u)e^{j2\pi uk/K}, k = 0, 1, 2, \dots, K-1$$

Fourier descriptors

- Boundary reconstruction using 546, 110, 56, 28, 14 and 8 Fourier descriptors out of a possible 1090.



Fourier descriptors

- This boundary consists of 64 point, P is the number of descriptors used in the reconstruction

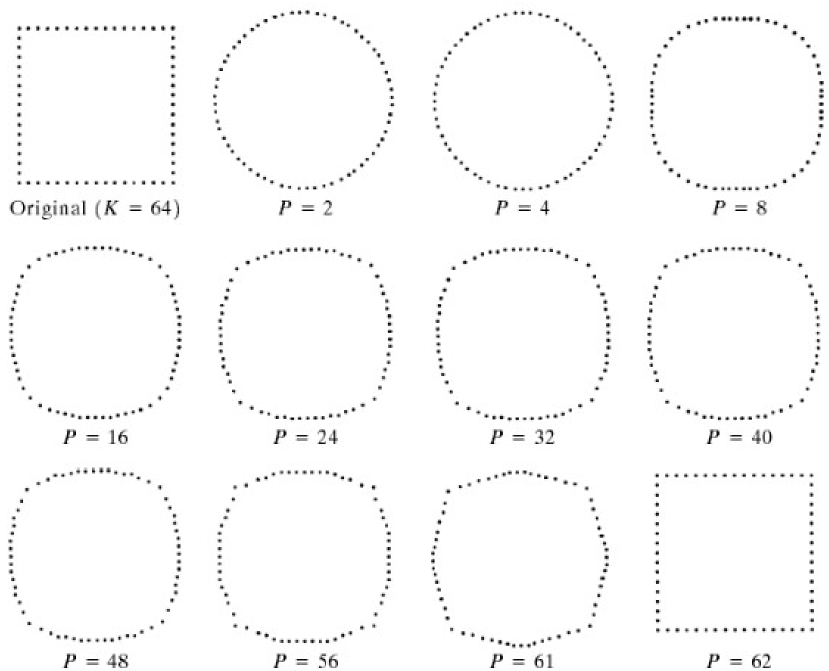


Image moments

- A particular weighted average of the image pixels' intensities
- Describe simple properties of a segmented image:
  - area (for binary images)
  - total intensity (for grayscale images)
  - centroid
  - orientation

Image moments

- Raw moments – for $p, q = 0,1,2, \ldots$ the raw moment M_{ij} is:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$$

- Area (or sum of gray intensities) = M_{00}
- Centroid $\{\bar{x}, \bar{y}\} = \left\{\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right\}$

- Central moments – for $p, q = 0,1,2, \ldots$:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

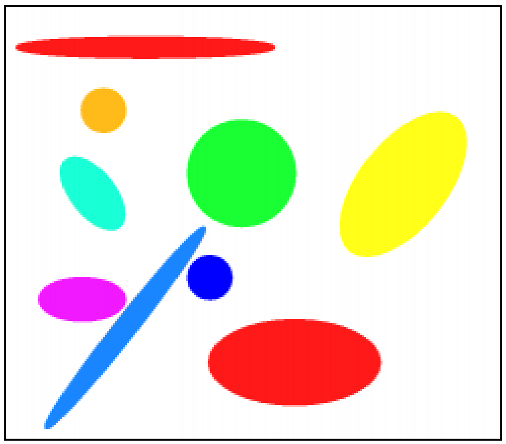
Simple Regional Descriptors

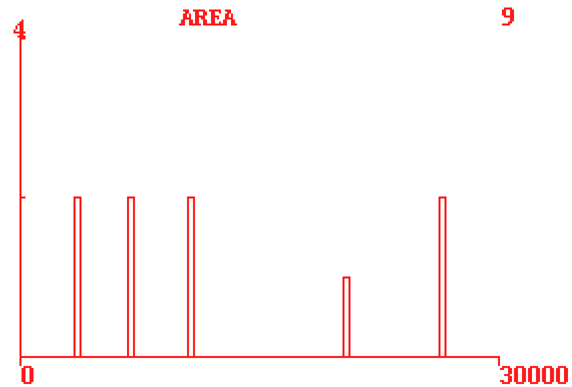
- Area = number of pixels in a region
- Compactness (P2A) = perimeter^2 / 4×π×area
- Circularity ratio = 4×π×area / perimeter^2 •

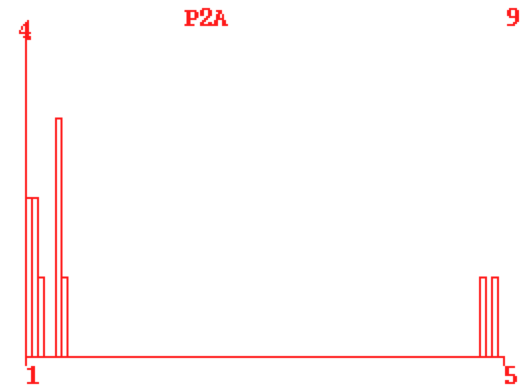
Graylevel measures

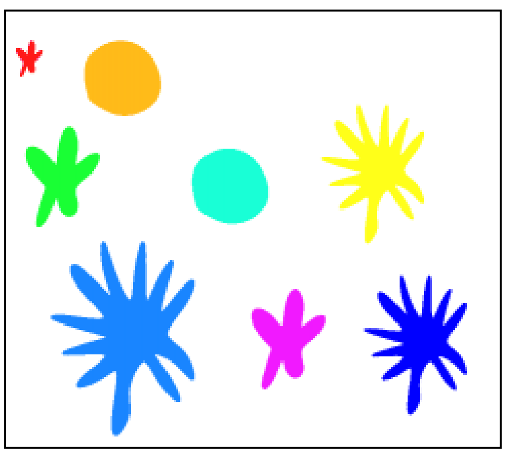
  - Mean
  - Median
  - Max
  - Etc.

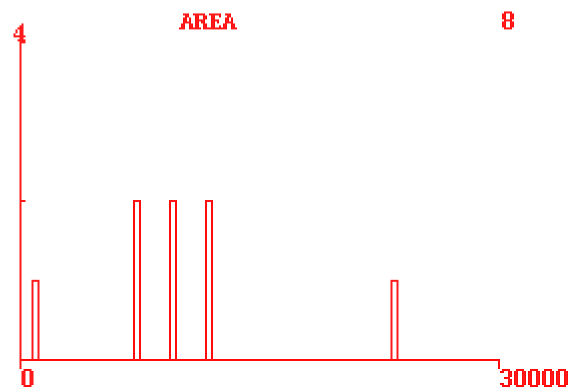
Examples of P2A vs area

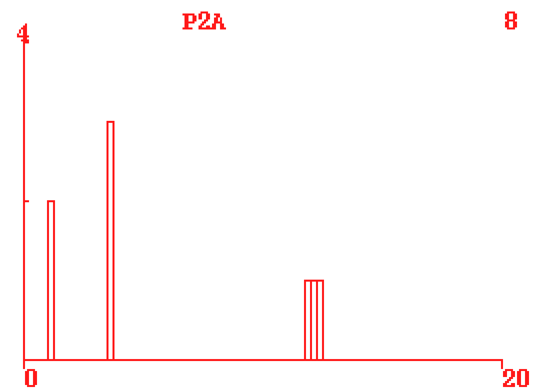








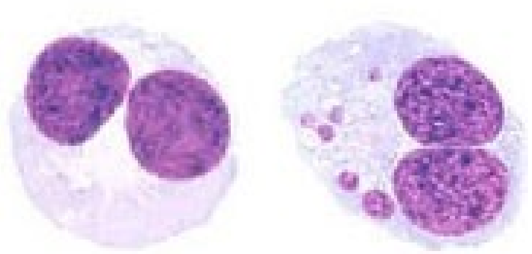




Topological descriptors

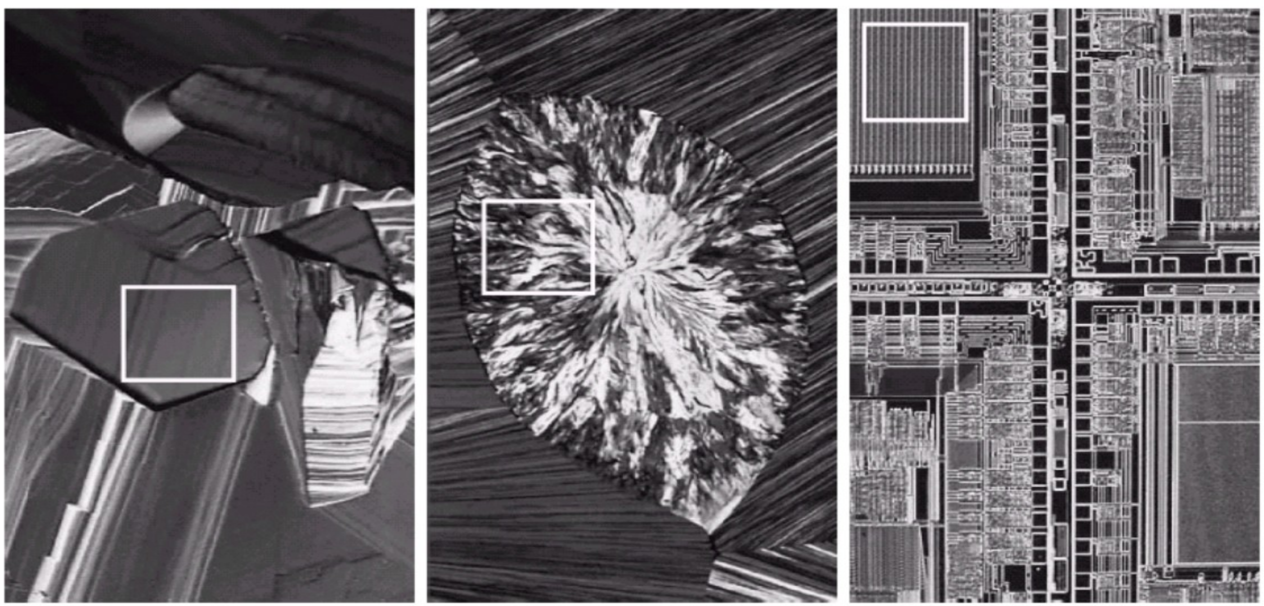
- Topology = The study of the properties of a figure that are unaffected by any deformation
- Topological descriptors
  - Number of holes in a region, H
  - Number of connected components, C
  - Euler number, E = C – H

A B C i o å ö 5 9 8



Texture

- Textures can be very valuable when describing objects
- Example below: Smooth, coarse and regular textures



Texture

- Statistical texture descriptors:
  - Histogram based
  - Co-occurence based

  (Statistical moments, Uniformity, entropy,...)

- Spectral texture descriptor
  - Use fourier transform

Histogram based descriptors

- Properties of the graylevel histogram, of an image or region, used when calculating statistical moments
  - z : discrete random variable representing discrete graylevels in the range [0 , L-1]
  - $P(z_i)$: normalized histogram component, i.e. the probability of finding a gray value corresponding to the i:th gray level z_i.

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i), \quad m = \sum_{i=0}^{L-1} z_i p(z_i)$$

2nd moment : Variance of z (contrast measure)

3rd moment : Skewness

4th moment : Relative flatness

Histogram based descriptors

Two **other** common histogram based texture measures:

- Uniformity (maximum for image with just one grayvalue):

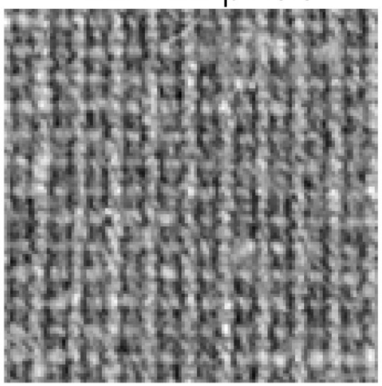
$$U = \sum_{i=0}^{L-1} p^2(z_i)$$

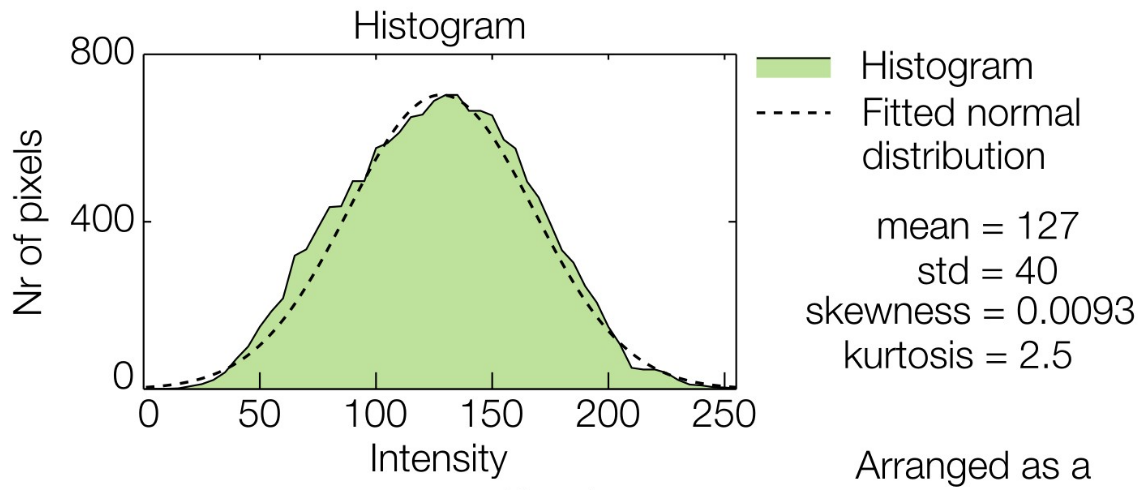
- Average entropy (measure of variability, 0 for constant images)

$$e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i)$$

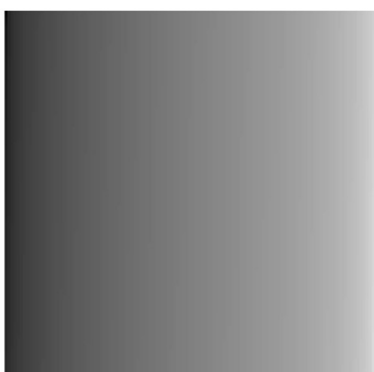
Intensity histogram says nothing about the spatial distribution of the pixel intensities

Original image
122x122 pixels

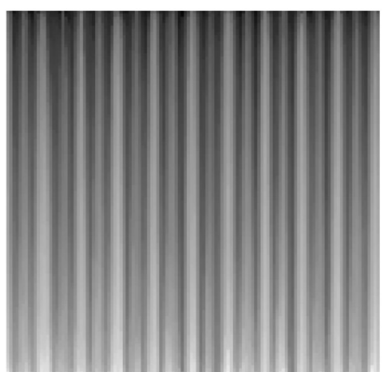




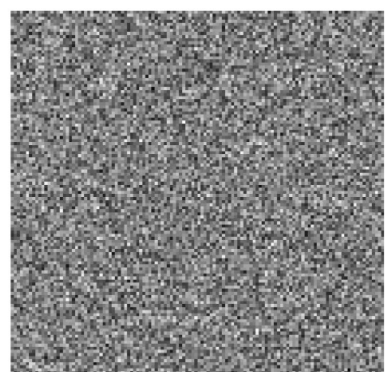
Pixels sorted



Rows sorted



Random permutations



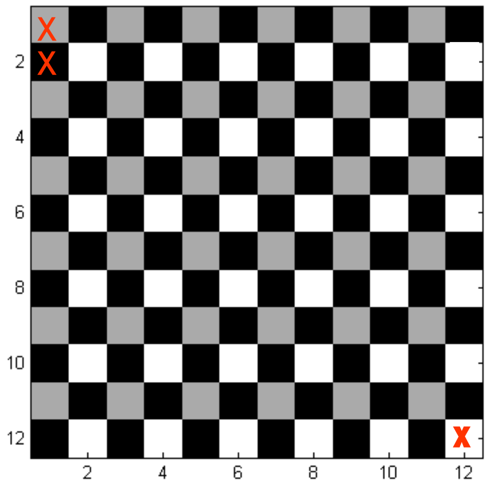
Arranged as a painting by René Magritte

Co-occurrence matrix

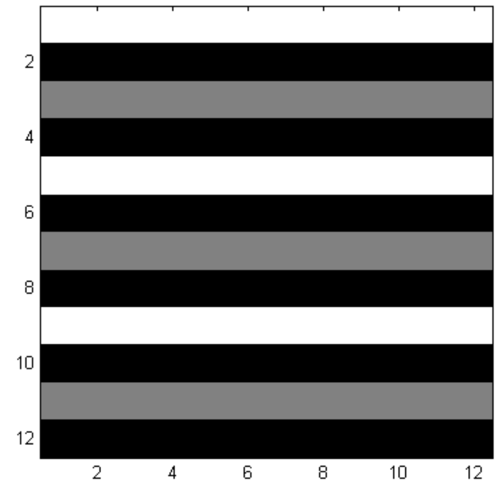
- For an image with N graylevels, and P, a positional operator, generate **A**, a N × N matrix, where $a_{i,j}$ is the number of times a pixel with graylevel value z_i is in relative position P to graylevel value z_j

- Divide all elements in **A** with the sum of all elements in **A.** This gives a new matrix **C** where $c_{i,j}$ is the probability that a pair of pixels fulfilling P has graylevel values z_i and z_j which is called the **co-occurrence matrix**

Building the matrix A



P=one pixel to the right



	0	1	2
0		30	36
1	36		
2	30		

What will the matrix look like for the striped image if P= one pixel down?

	0	1	2
0	66		
1		33	
2			33

Co-occurrence matrix Descriptors

- **Maximum probability (strongest response to P)** $max_{i,j}(c_{ij})$

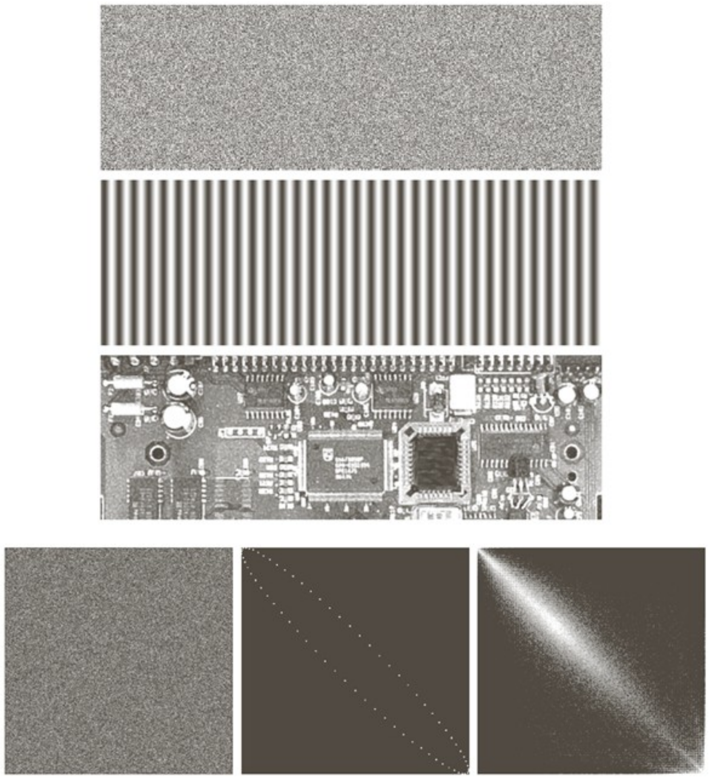
- **Uniformity** $$\sum_i \sum_j c_{ij}^2$$

- **Entropy (randomness)** $$-\sum_i \sum_j c_{ij} \log_2 c_{ij}$$

How can rotation robust measures be achieved?

Co-occurrence matrix

- Match image with a co-occurrence matrix!



max prob:	0.00006	0.01500	0.0680
Uniformity:	0.00002	0.01230	0.00480
Entropy:	15.75	6.43	13.58

How to choose / design representations and descriptors:

- Find/create representations/descriptors that are invariant to transformations that are unimportant for your task:
  - e.g. noise, scale, blur, …
- Find/create representations and descriptors that are relevant for your question
  - height, to classify adults / children
  - color and shape to separate bananas, pears and tomatoes
- Be creative
- Stay as simple as possible