Binary Object Representations

Lecture IPIU
Thomas Breuel

why?

- **reduce amount of data**
- **focus on relevant attributes**
- **remove noise / irrelevant attributes**
- **treat recognition like database lookup:**
  - “The object is bilaterally symmetric, has one hole, one “v” junction and two tips.” — “A”

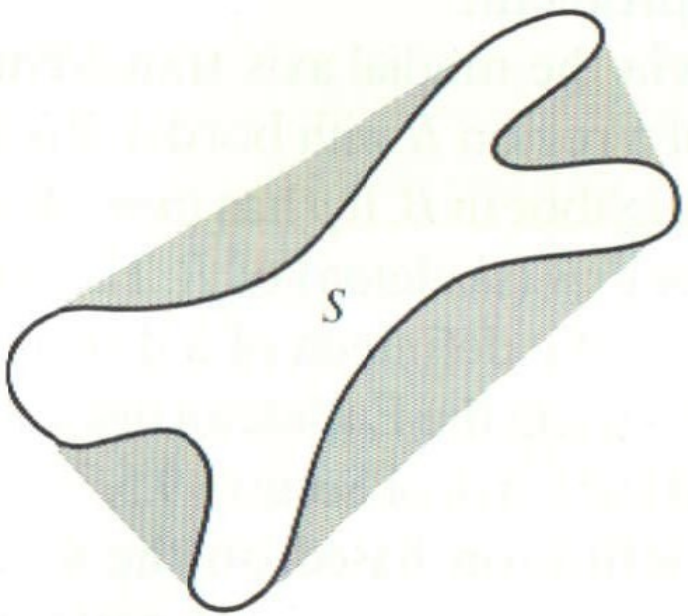
moments

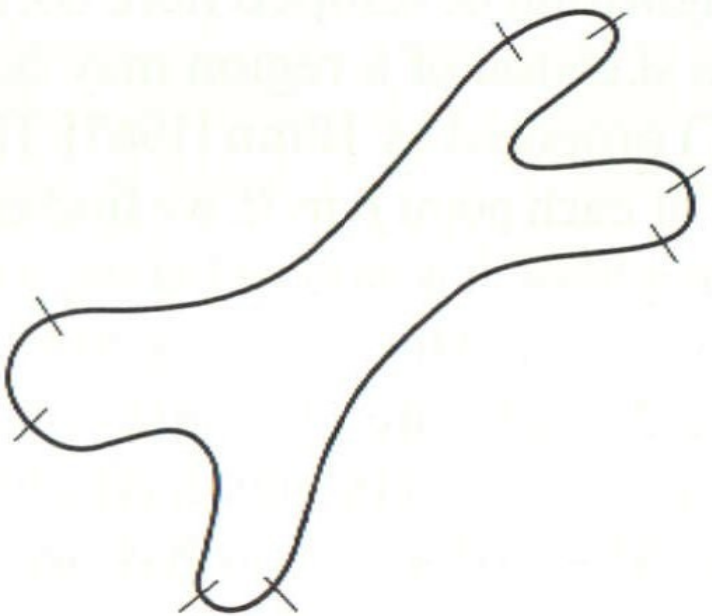
- **answer the questions:**
  - where is the object?
  - how big is it?
  - how is it oriented?
- **(worksheet)**

other simple descriptors

- area (# pixels)
- boundary length (# border pixels)
- compactness
- bounding box
- aspect ratio of bounding box
- aspect ratio of major/minor axes
- amount of convex deficiency

convex deficiency



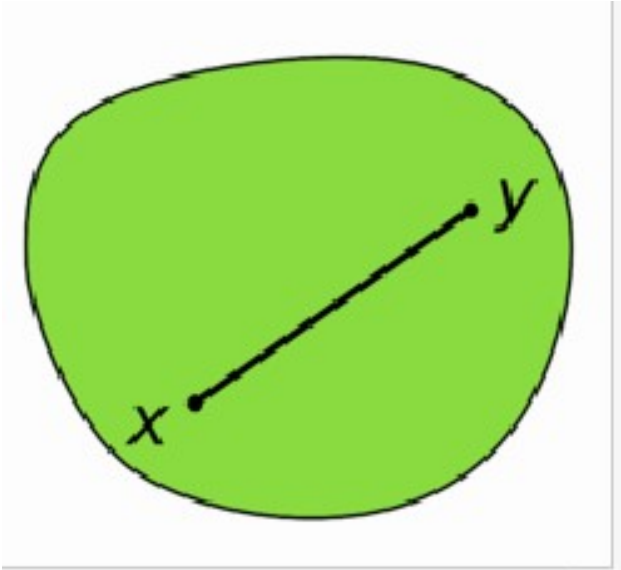


difference between convex hull and shape

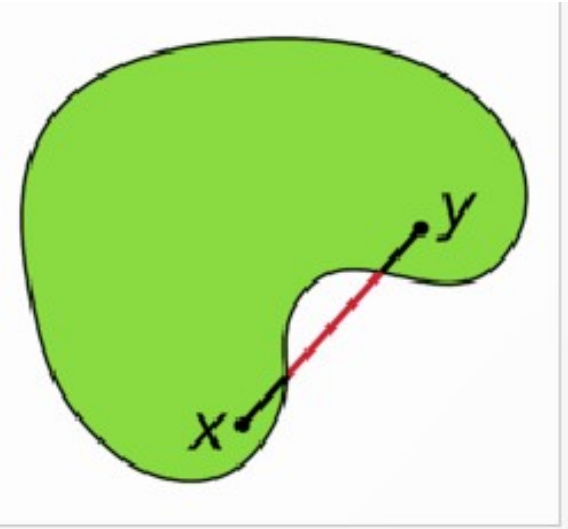
convex hull = the (!) minimal convex set containing all the points

convex set = ?

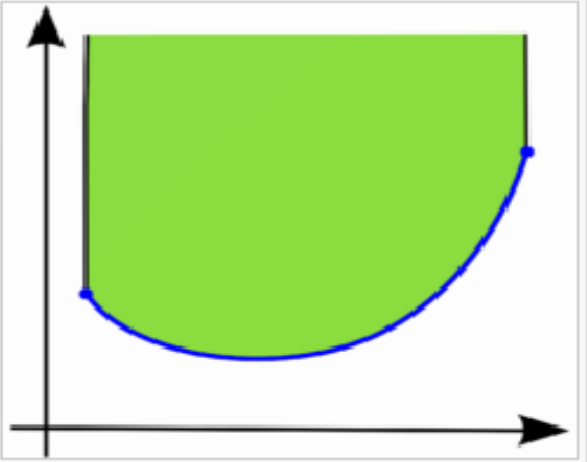
convex set



A convex set.



A non-convex set, with a line-segment outside the set.

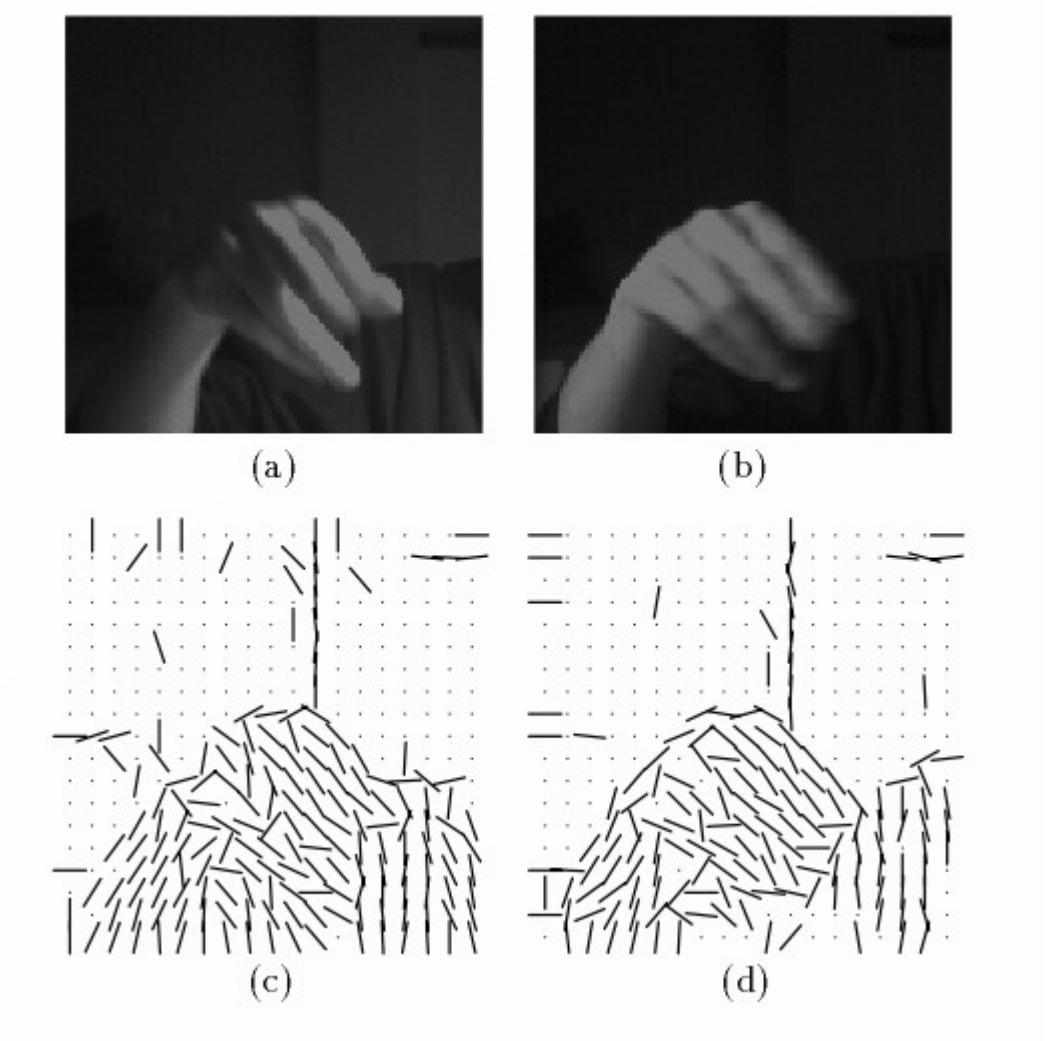


A function is convex if and only if the region (in green) above its graph (in blue) is a convex set.

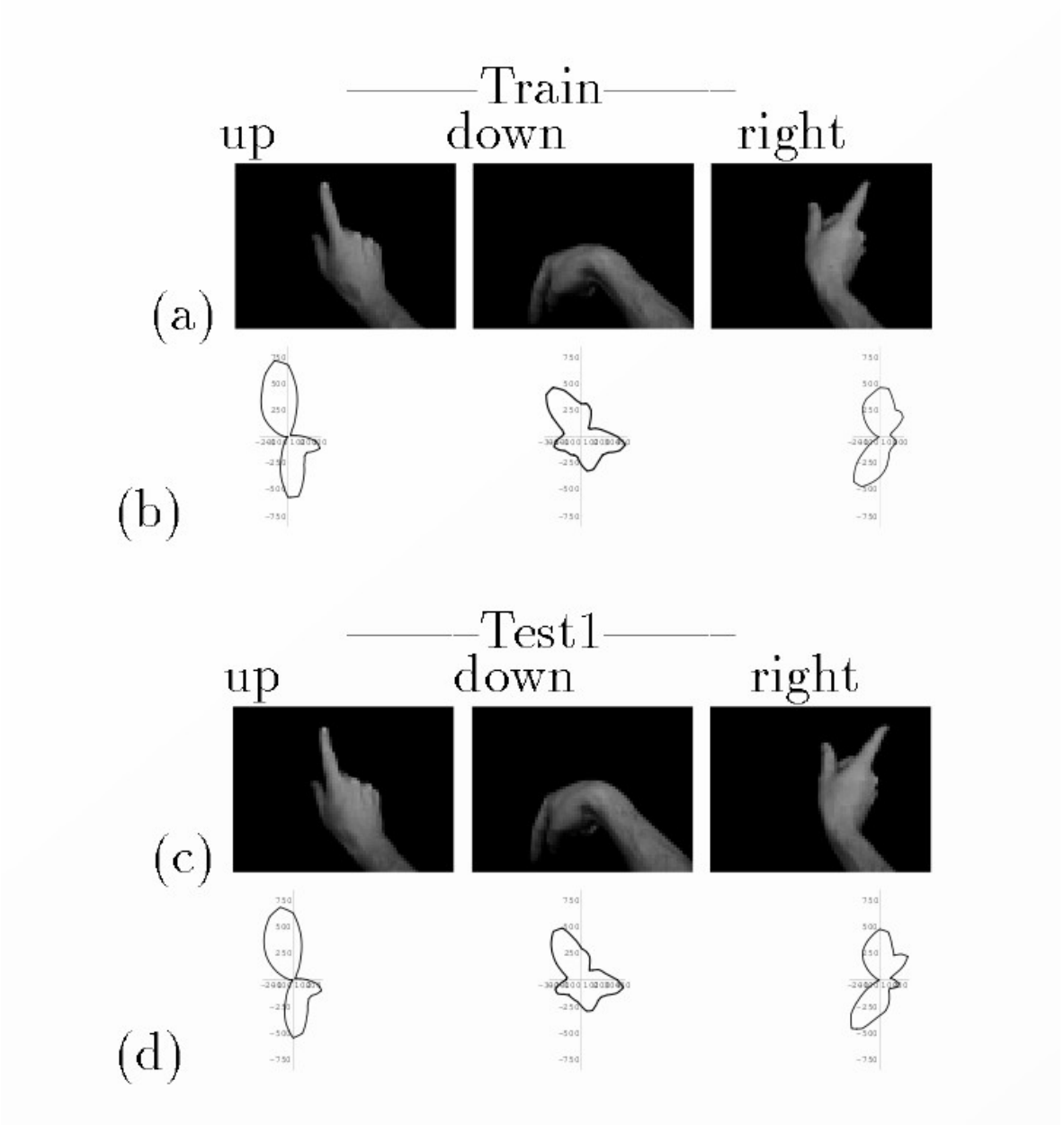
topological descriptors

- # holes
- # endpoints
- # T-junctions
- # X-junctions
- # convex deficiencies

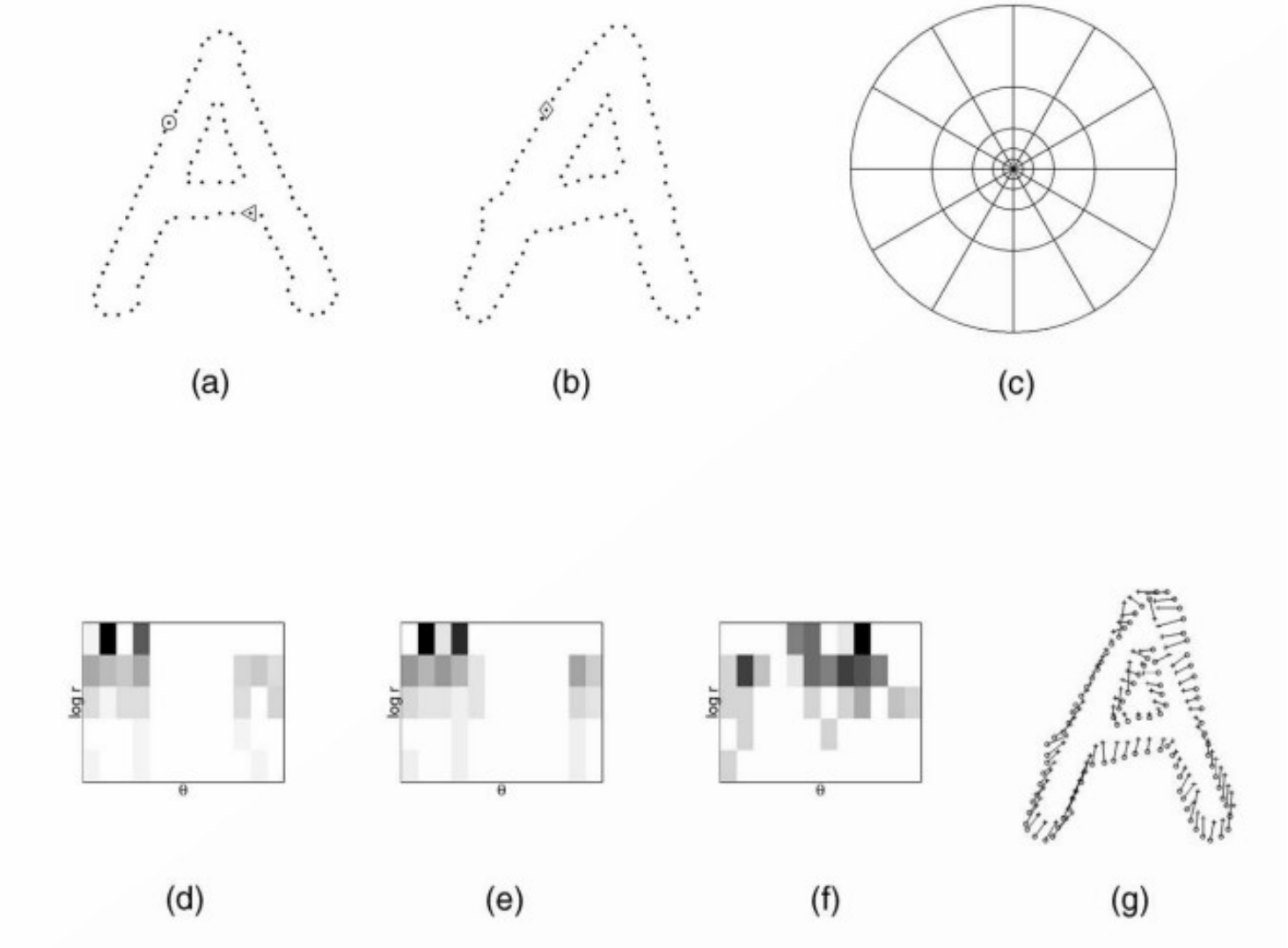
orientation histograms



orientation histograms



shape context

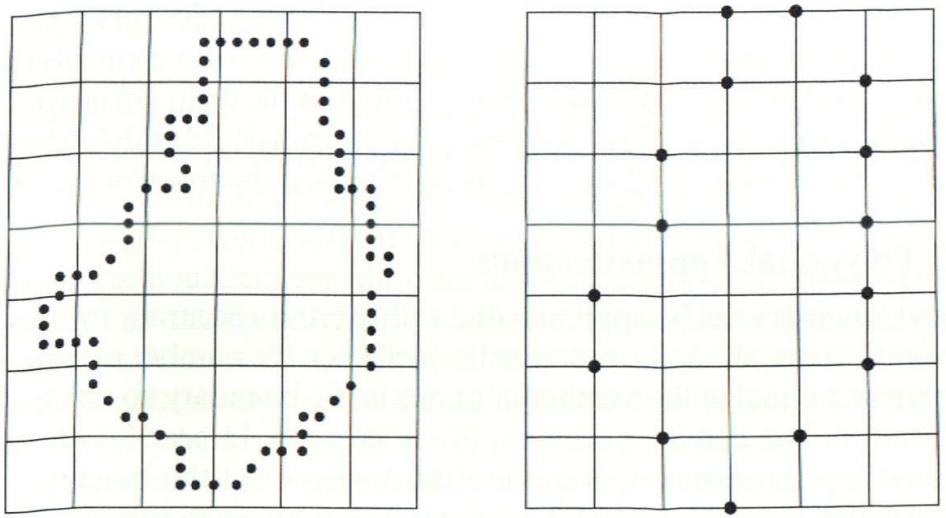


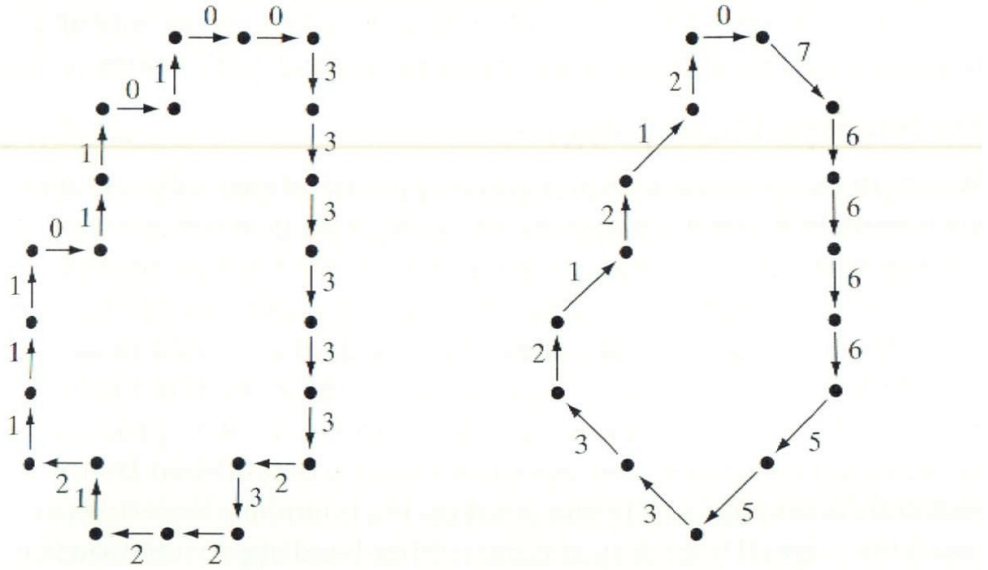
CHAIN CODES

chain codes

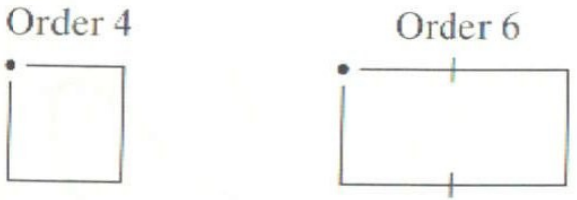
- **reduce shapes to character strings**
- **use string techniques for shape recognition**
  - syntactic pattern recognition
  - grammars
  - string edit distance

chain codes

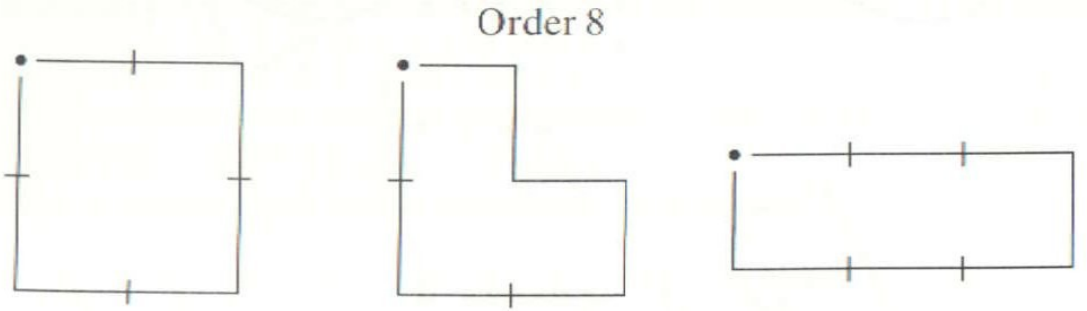




differences, shape numbers



Chain code:	0 3 2 1	0 0 3 2 2 1
Difference:	3 3 3 3	3 0 3 3 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3



Chain code:	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Difference:	3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0	3 0 0 3 3 0 0 3
Shape no.:	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3

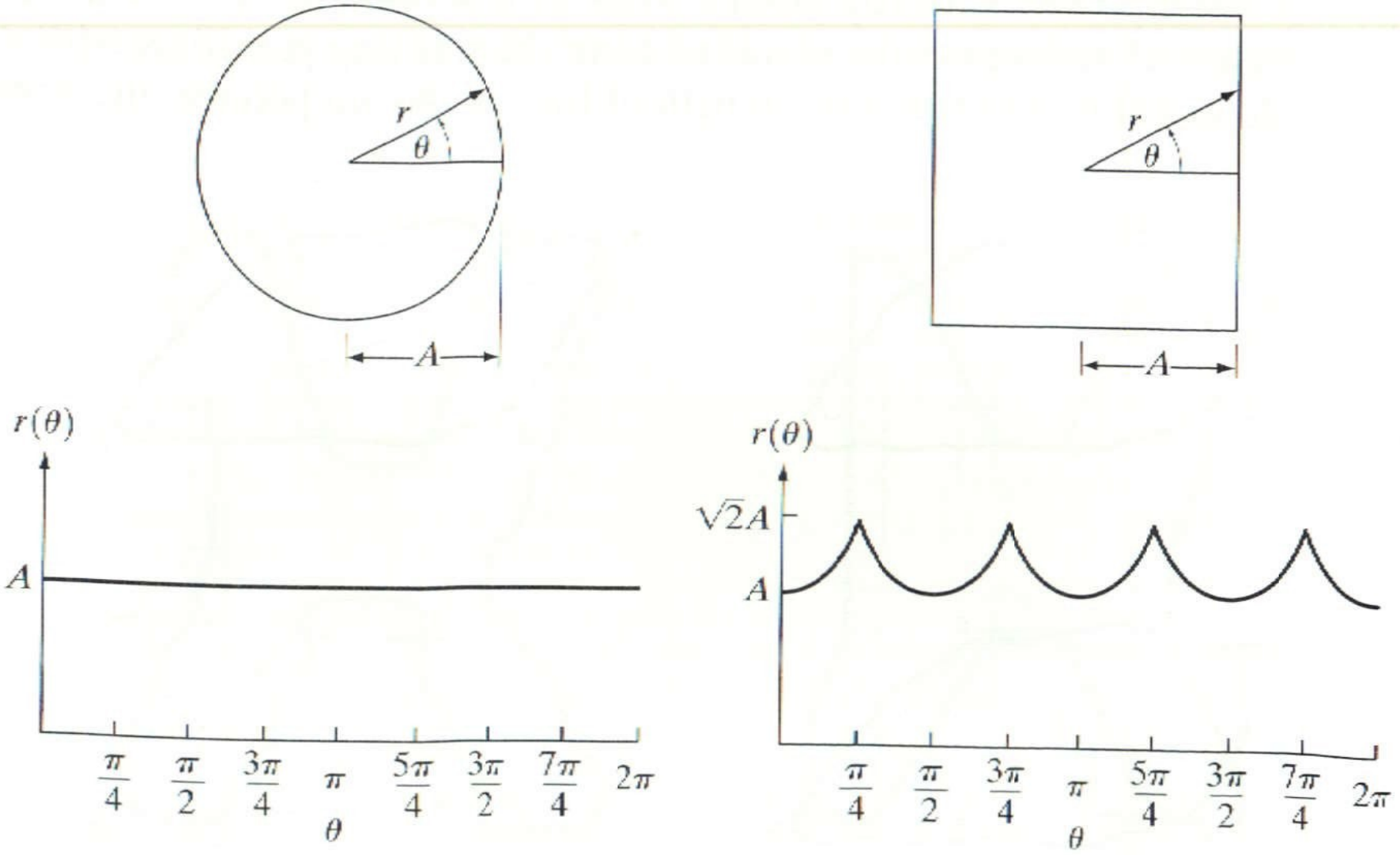
chain codes

- transform 2D shapes into 1D strings
- using differences makes sequence rotation independent
- using shape numbers makes sequence independent of starting point

- so why isn't everybody using this?

CURVE REPRESENTATIONS

shape signatures



mathematical curves

mathematical curve

$$f:[0,1]\rightarrow R^2$$

regular curve

$$\frac{\partial f}{\partial x}\neq 0 \quad \vee \quad \frac{\partial f}{\partial y}\neq 0$$ everywhere

natural parameterization (by arclength)

$$f:[0,l]\rightarrow R^2$$

closed curve

$$f(0)=f(l)$$

complex numbers

$$f : [0, l] \rightarrow C$$

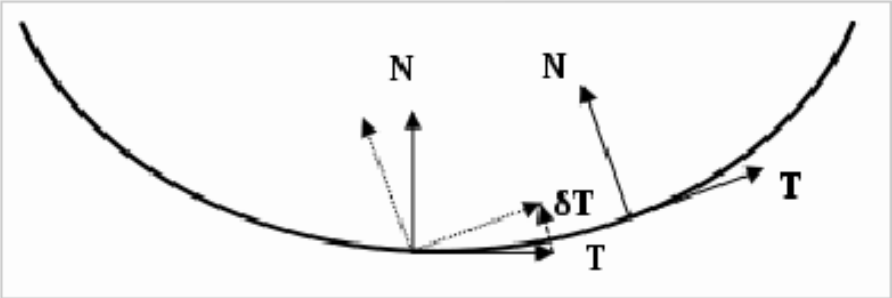
- **consider curve a complex-valued function**
  - translation → addition of complex number
  - rotation → complex multiplication by constant
  - starting point → translation of periodic function
    = multiplication in Fourier domain

curvature

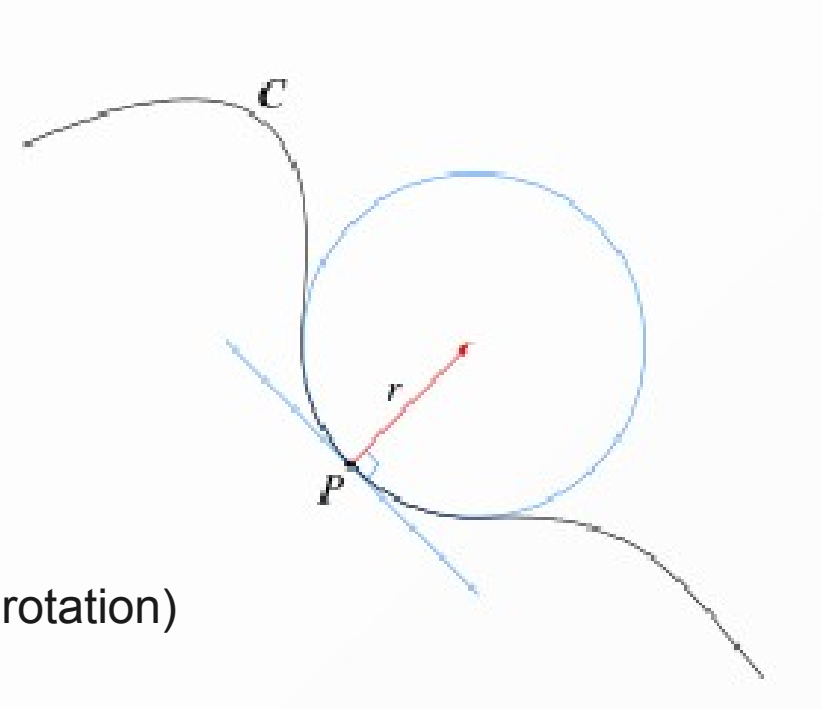
gamma: naturally param. curve, T: tangent (vector that's function of s)

kappa: curvature, R: radius of curvature

$$T(s) = \gamma'(s), \quad T'(s) = \kappa(s)N(s), \quad \kappa(s) = \|\gamma''(s)\| = |k(s)|, \quad R(s) = \frac{1}{\kappa(s)}.$$



The **T** and **N** vectors at two points on a plane curve, a translated version of the second frame (dotted), and the change in **T**: δ**T**'. δs is the distance between the points. In the limit $\frac{d\mathbf{T}}{ds}$ will be in the direction **N** and the curvature describes the speed of rotation of the frame.



(signed curvature: sign determined by direction of rotation)

curvature

- **invariances**
  - tangent direction — translation
  - curvature — translation + rotation
  - still have starting point uncertainty
- **idea**
  - transform input image into curvature-vs-arclength
  - match curvature-vs-arclength representation
- **what do you think?**