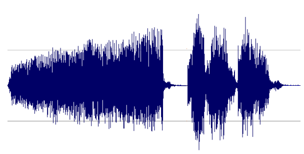
Computing with Shape

Ralf Juengling
10/8/2007 Portland State University

Sensor data is non-discrete

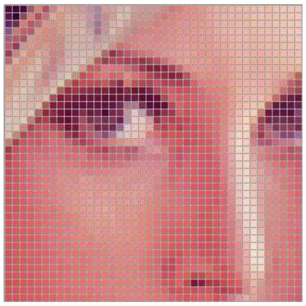
- Sound
- Images
- Video



Signals are discretized by recording devices (quantization & sampling). But conceptually the data are still continuous functions.

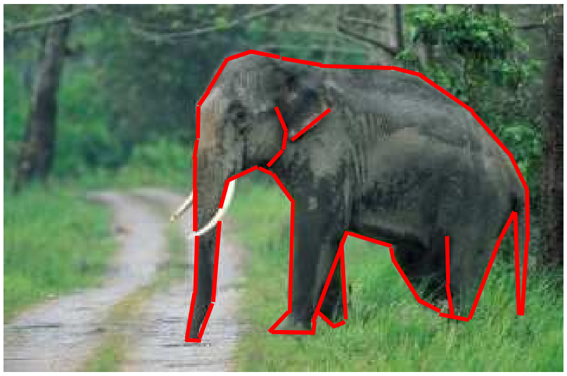
Example: Images

- Pixel representation good for storage and reproduction
- Not good for finding objects in images



For solving image understanding problems you will in general need to map an image to something more apt to the task.

If we could segment objects…



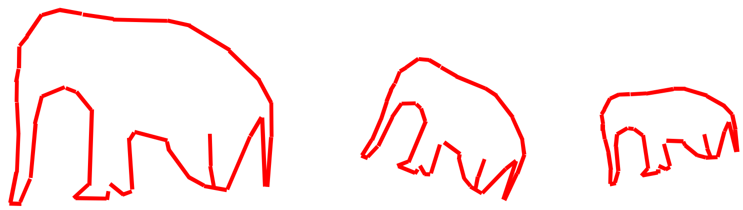
(This is an unsolved problem in general.)

… we would get an object's silhouette

Difference between **silhouette** and **shape**?

Silhouette = a particular closed curve (a polygon)

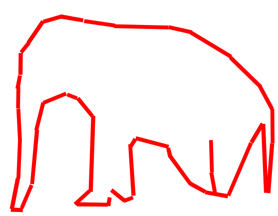
Shape = all geometrically similar silhouettes



Two curves are *geometrically similar* if one may be mapped onto the other by a combination of *translation*, *rotation* and *scaling*.

Computational problems with silhouettes

- *Shape of a silhouette*
  Are these two silhouettes of the same shape?
- *Silhouette classification*
  Is this silhouette of a car, a telephone, or an elephant?
- *Understanding spatial arrangement*
  Which way is the head facing?
- *Description of special features*
  How long is the trunk?

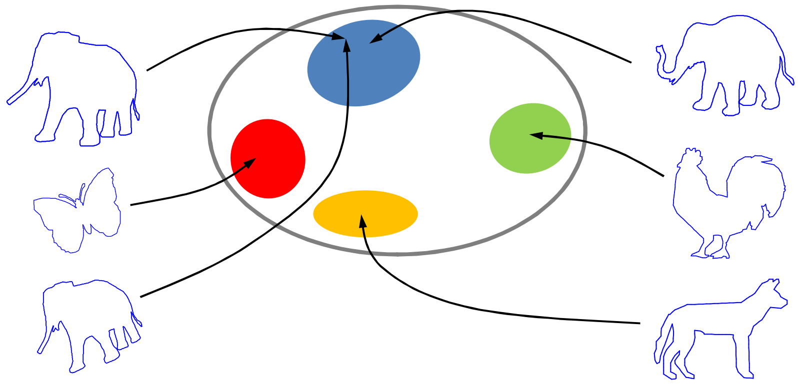


Using an appropriate silhouette representation makes solving these problems much easier!

Silhouette Classification

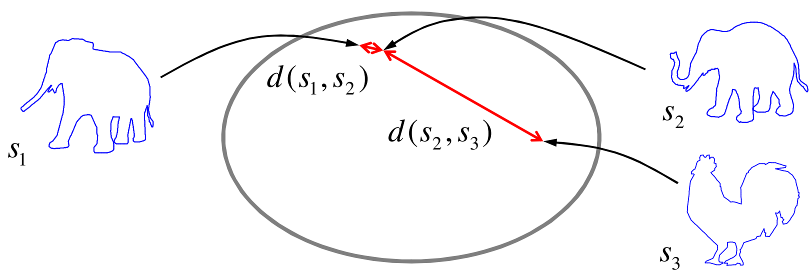
Typical approach:

- Map shapes to a feature space and train a classifier (partition the space)



Silhouette Comparison

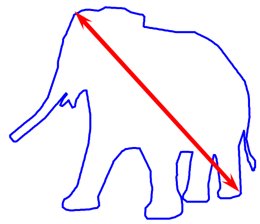
- Shape features also used in comparing silhouettes by means of a *distance function*



Shape features

Some simple features:

- *Area*
- $Perimeter^2 / Area$
- $Area / Area\,(Convex\,Hull)$
- *Length(Diagonal)*
- *Orientation(Diagonal)*



Which of these features are invariant to:

- Translation?
- Rotation?
- Scaling?

Fourier Descriptors

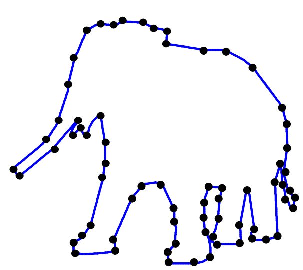
- Popular way of generating a large number of invariant shape features
- Development here for polygonal closed curves due to Zahn & Roskies (1972)

Silhouette represented by coordinate functions

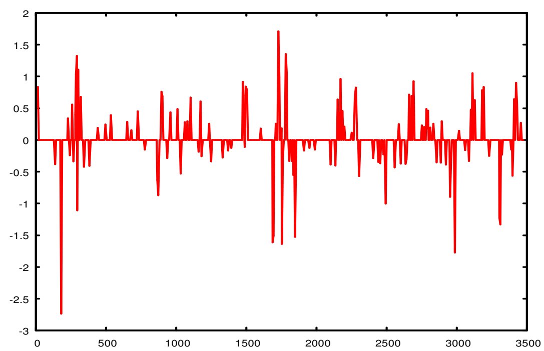
$x(l), y(l)$ with $0 \le l \le L$, and

$x(0) = x(L)$, $y(0) = y(L)$.

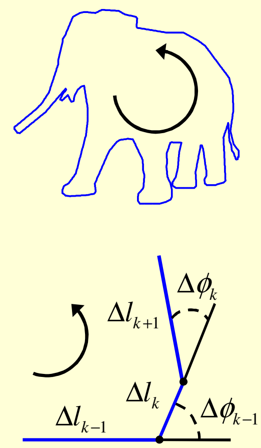
The first step is to switch to another representation!



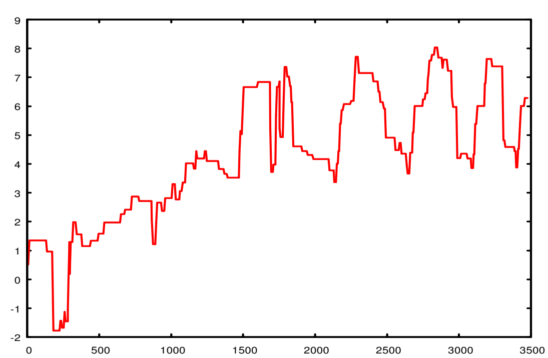
1. Turning angle over arc-length



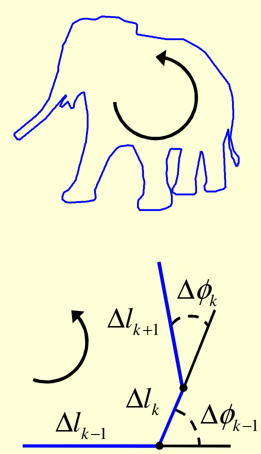
$$\phi'(l) = \begin{cases} \Delta\phi_k & \text{if } l = \sum_{i=1}^{k} \Delta l_i \\ 0 & \text{otherwise} \end{cases}$$



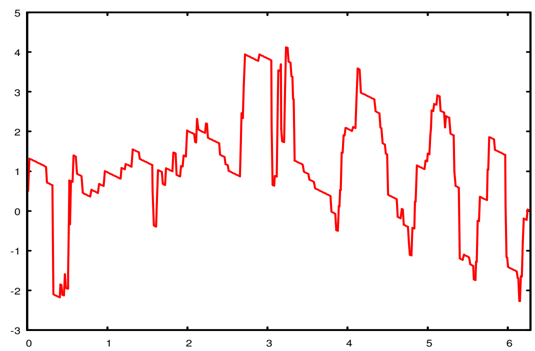
2. Turn into cumulative turning angle...



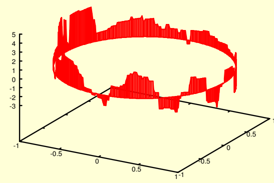
$$\phi(l) = \sum_{i=1}^{k} \Delta\phi_i \,,\ \sum_{i=1}^{k} \Delta l_i \le l \le \sum_{i=1}^{k+1} \Delta l_i$$



3. … and into a periodic function on [0, 2π]



$$\phi^*(t) = \phi(Lt/2\pi) - t\,,\ 0 \le t \le 2\pi$$

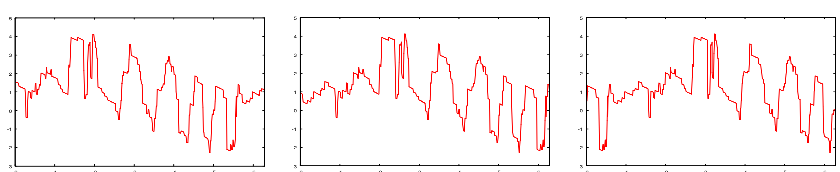


$$\phi^*(0) = \phi^*(2\pi)\ !$$

ϕ^* is (almost) a representation of shape

We can use ϕ^* to create a silhouette that looks like the original. Note that we shed information regarding size, orientation and location.

But ϕ^* depends on choice of starting point on the curve!



One more step!

4. Write ϕ^* as a Fourier series

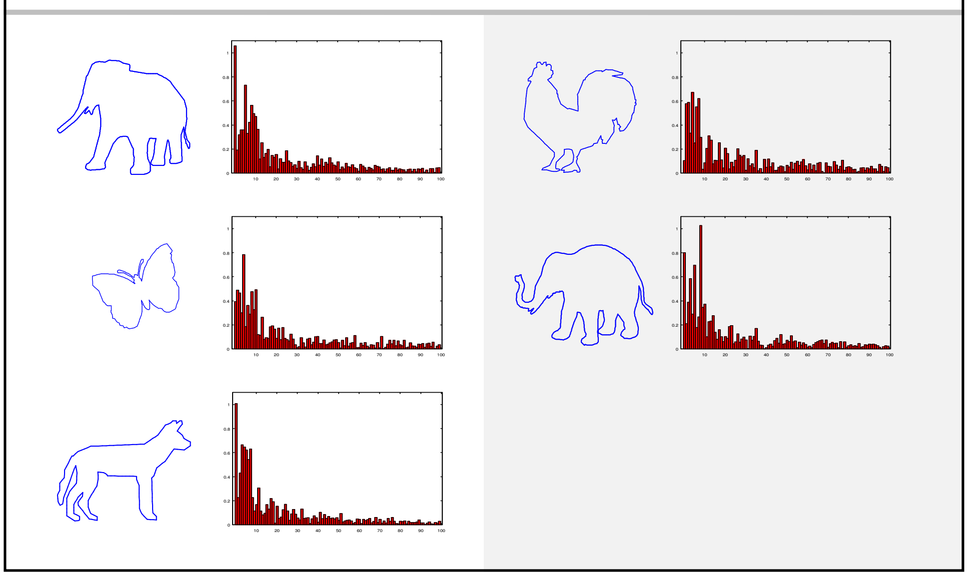
$$\phi^*(t) = \mu_0 + \sum_{k=1}^{\infty}\left(a_k \cos kt + b_k \sin kt\right)$$

$$= \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k)$$

$\{A_k\}$ are called *amplitudes*, $\{\alpha_k\}$ are called *phases*.

- We are interested in the amplitudes because they don't change when ϕ^* is shifted circularly!
- We take finite # of coefficients and approximate ϕ^*
  $\phi^*(t) \approx \mu_0 + \sum_{k=1}^{N} A_k \cos(kt - \alpha_k)$
- $\{A_k\}_{k=1\ldots N}$ are pure shape features!

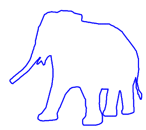
Examples (N=100)



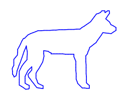
A simple distance function

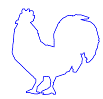
Let **A** and **B** be vectors of amplitude values.
A very simple distance function is the Euclidean norm:

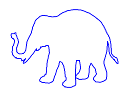
$$d(\mathbf{A},\mathbf{B}) = \|\mathbf{A}-\mathbf{B}\|_2 = \sqrt{\sum_{i=1}^{N}\left(A_i - B_i\right)^2}$$

Distance of  to :


1.144

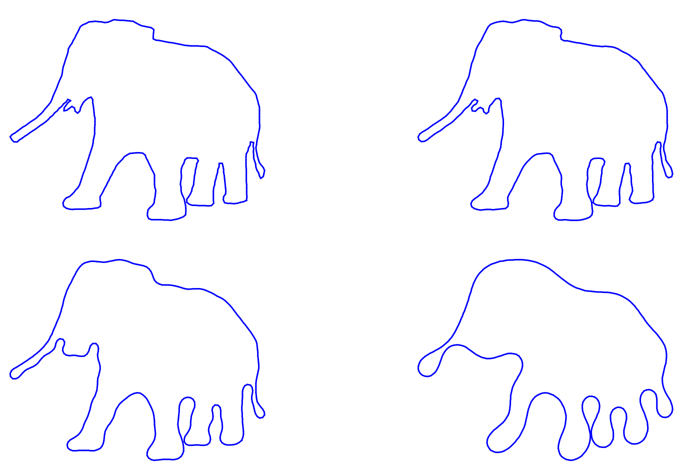

0.908


1.508


0.837

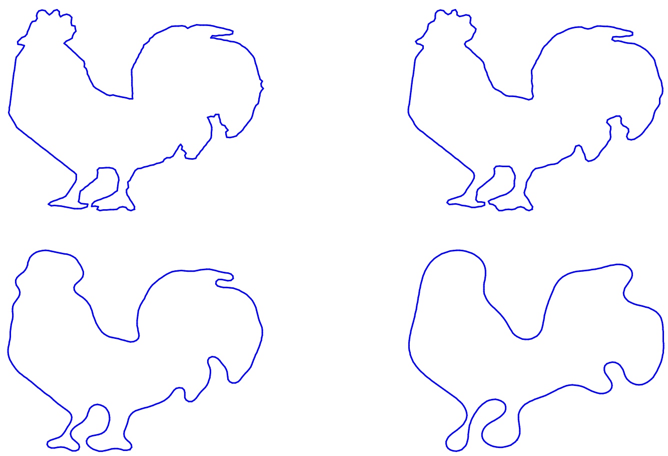
How many numbers $\{A_k\}$ make sense?

Reconstructions with N=200, 100, 50, 20:



How many numbers $\{A_k\}$ make sense?

Reconstructions with N=200, 100, 50, 20:



Computing the Fourier descriptors

$$\phi^*(t) = \mu_0 + \sum_{n=1}^{\infty} A_n \cos(nt - \alpha_n)$$

$$a_n = -\frac{1}{n\pi}\sum_{k=1}^{m} \Delta\phi_k \sin nt_k$$

$$b_n = \frac{1}{n\pi}\sum_{k=1}^{m} \Delta\phi_k \cos nt_k$$

$$A_n = \sqrt{a_n^2 + b_n^2}$$

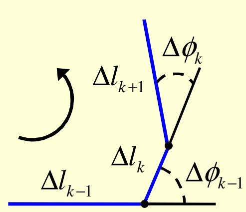
$$\alpha_n = \operatorname{atan2}(b_n, a_n)$$

$m \mathrel{\hat=} \#$polygon vertices

$L \mathrel{\hat=}$ polygon perimeter

$l_k = \sum_{i=1}^{k} \Delta l_i$

$t_k = 2\pi l_k / L$



Fourier Descriptors Summary

- Relatively simple to implement
- FD amplitudes in order of increasing detail
- Feature format allow for simple distance functions…
- … and the use of "standard classifiers" (SVM, NN,…)
- However, FDs do not reveal perceptually important shape features

References:

C. T. Zahn and R. S. Roskies: "Fourier descriptors for plane closed curves". *IEEE Transaction on Computers,* C-21(3):269—281, 1972.

R. N. Bracewell: *The Fourier Transform and its Applications.* McGraw-Hill, New York, 2nd edition, 1986.

Morphological shape features

Other shape features in terms of "components":

- Number of components
- Relative size of components
- Spatial arrangements of components

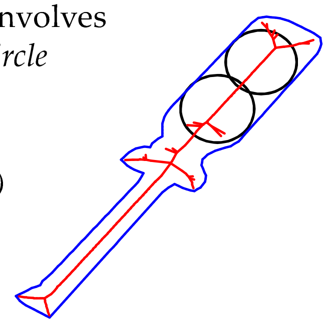
Problem:

How to determine "components" or "parts"?

Introducing: Shape skeleton

Important tool for morphological analysis

- Each point on the skeleton corresponds to two or more boundary points in *local symmetry*
- Definition proposed by Blum involves concept of *maximum inscribed circle*

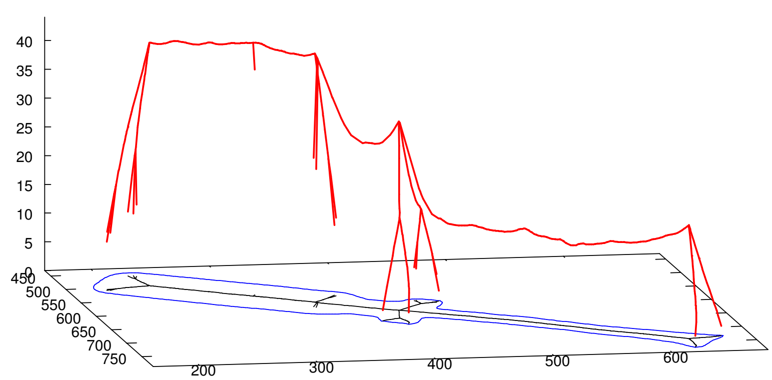


Goes under various names:

- *Symmetric Axis Transform (SAT)*
- *Medial axis transform (MAT)*
- *Shock graphs*

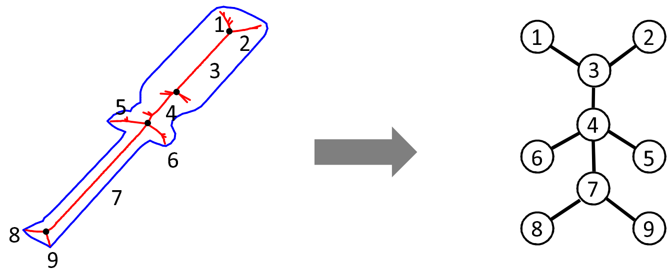
SAT is a shape representation!

If you also record the radius of the maximum inscribed circles , you may reproduce the silhouette.



SAT for morphological analysis

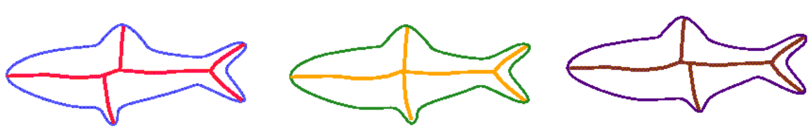
- Break skeleton into pieces at bifurcation points
- Compute attributes for every branch of the skeleton
- Map skeleton to an attributed graph



B. Blum and R. N. Nagel: Shape Description using weighted Symmetric Axis Features. *Pattern Recognition* 10(3):167—180, 1978.

Small deformations may change graph

A problem with this approach: skeleton bifurcations are sensitive to small changes in silhouette!



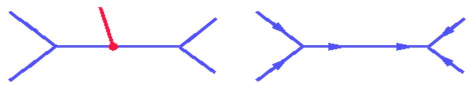
We would get three different graphs here!

(Figs. by Thomas Sebastian, http://www.lems.brown.edu/~tbs/Talk2.ppt)

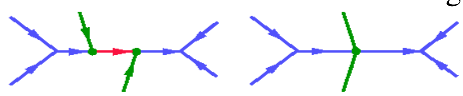
One solution: Edit distance

Take possible changes into account in distance function:

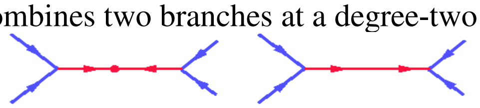
Splice: deletes a shock branch and merges the remaining two



Contract: deletes a shock branch between degree-three nodes



Merge: combines two branches at a degree-two node



Deform: changes the attributes of a shock branch

One solution: Edit distance (cont'd)

- Assign a cost to each edit operation
- Distance between two shape graphs: minimum-cost sequence of edits transforming one graph into other

Pros & Cons:

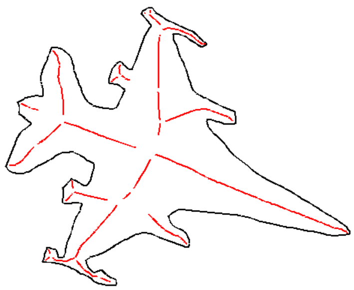
- Results for shape matching very good
- Difficult to implement
- Distance function expensive

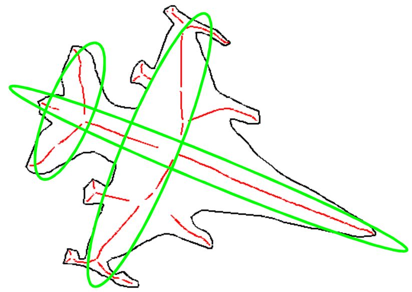
T. B. Sebastian, P. N. Klein and B. B. Kimia: Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 26(5):550--571, 2004.

Other solution: Use other components

Define components less sensitive to skeleton bifurcation

- Don't look at skeleton bifurcations
- **Group** skeleton fragments by "good continuation"

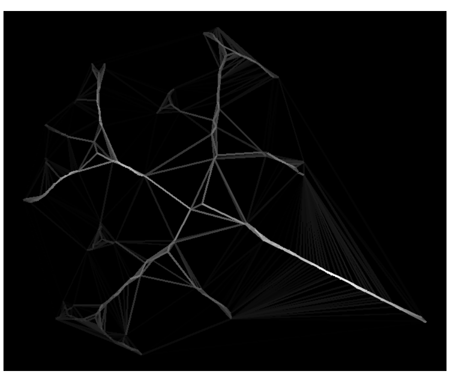


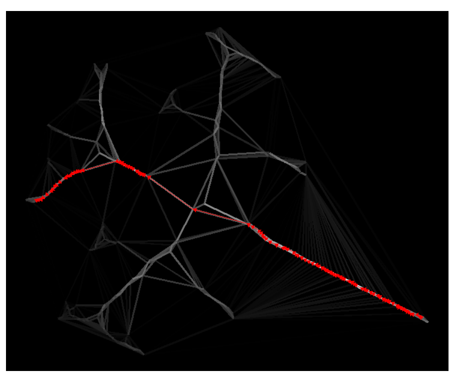


R. Juengling and L. Prasad: Parsing silhouettes without boundary curvature. *Proceedings of the 15th International Conference on Image Analysis and Processing, 2007.*

Grouping skeleton fragments

Grouping done with Sashua & Ullman's algorithm

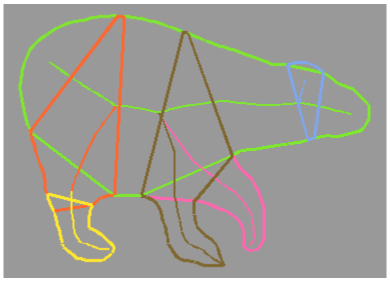
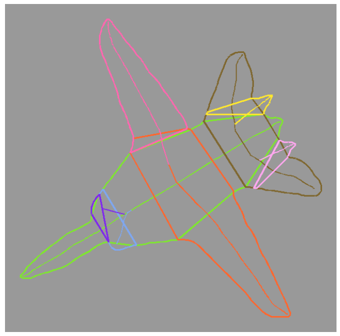




A. Shashua and S. Ullman: Structural Saliency: the Detection of Globally Salient Structures Using a Locally Connected Network. *In Proc. of the 2nd International Conference on Computer Vision*, 1988.

Components obtained by grouping

- Parts less sensitive to skeleton bifurcations
- Needs relatively "clean" skeleton (*skeleton pruning*)
- Method has yet to be tested for shape matching

Skeleton-based methods summary

- Parts may correspond to perceptual shape features
- Allows for different degrees of abstraction
- SAT representation supports solving other problems related to shape (alignment, feature measurement)
- Robust matching of different shapes instances (i.e., robustness w.r.t. intra-class variability)

- SAT not easy to compute
- Shape features don't have a simple format (graphs)
- Matching based on skeleton-derived features tend to be expensive