Representation and Detection of Shapes in Images

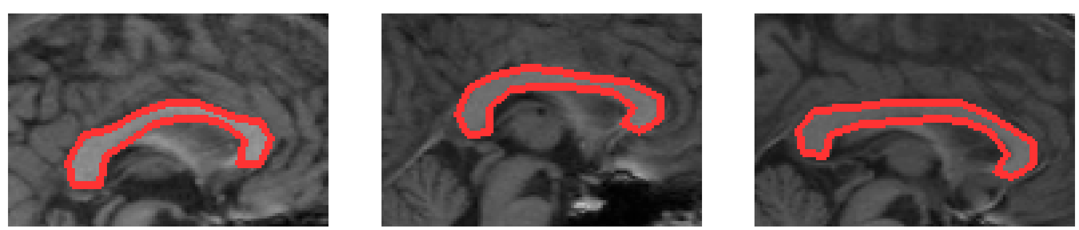
Pedro F. Felzenszwalb
Department of Computer Science
University of Chicago

Introduction

Study of shape is a recurring theme in computer vision.

- It is important for object recognition.

- Useful for model-based segmentation.



Talk outline

1. Representation of objects using triangulated polygons.

2. Finding a non-rigid object in an image.

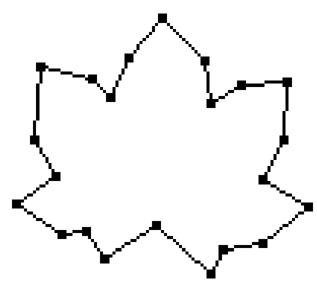
3. Learning a non-rigid shape model from examples.

4. Shape grammar for modeling generic objects.

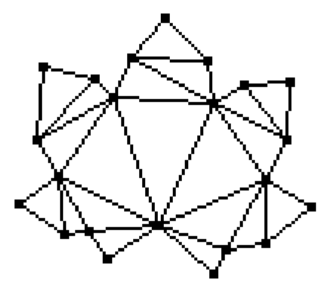
Triangulated polygon representation

- Consider two-dimensional objects with piecewise-smooth boundaries and no holes.
- Approximate object using a simple polygon P.
- A triangulation is a decomposition of P into triangles defined by non-crossing line segments connecting vertices of P.

object

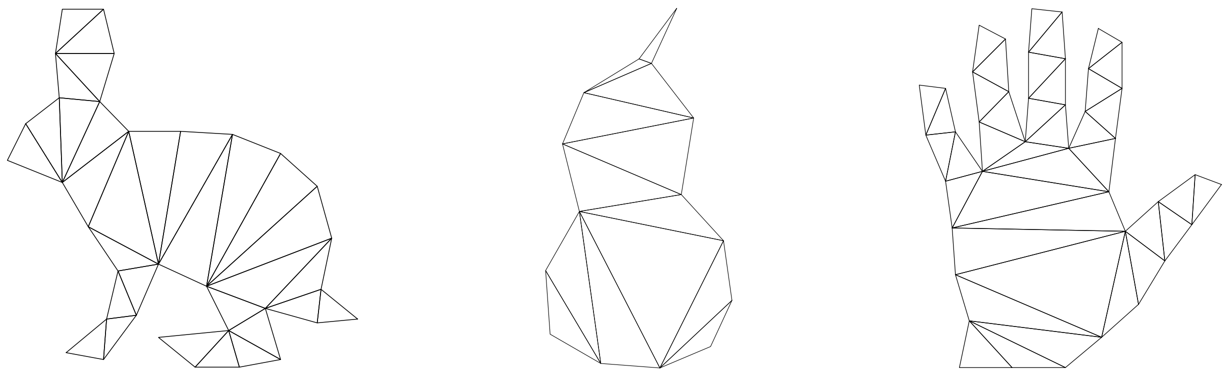


polygon



triangulation

Constrained Delauney triangulation

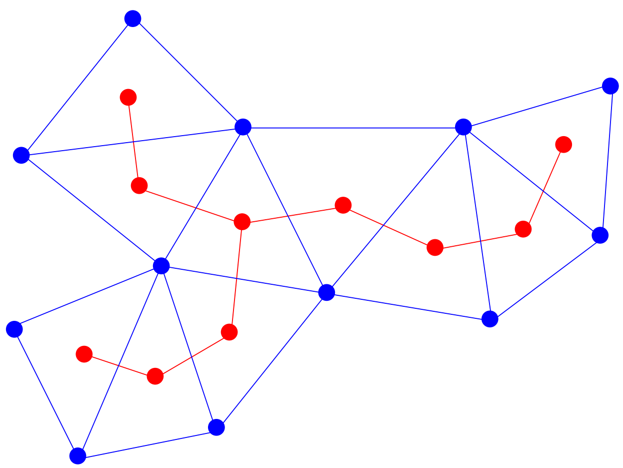


Natural decomposition of object into parts, closely related to the medial axis transform.

Definition. *The* constrained Delauney triangulation *contains the edge ab if a is visible to b and there is a circle through a and b that contains no vertex c visible to ab.*

Structural properties

There are two graphs associated with a triangulated polygon.



Dual graph of triangulated simple polygon is a tree.

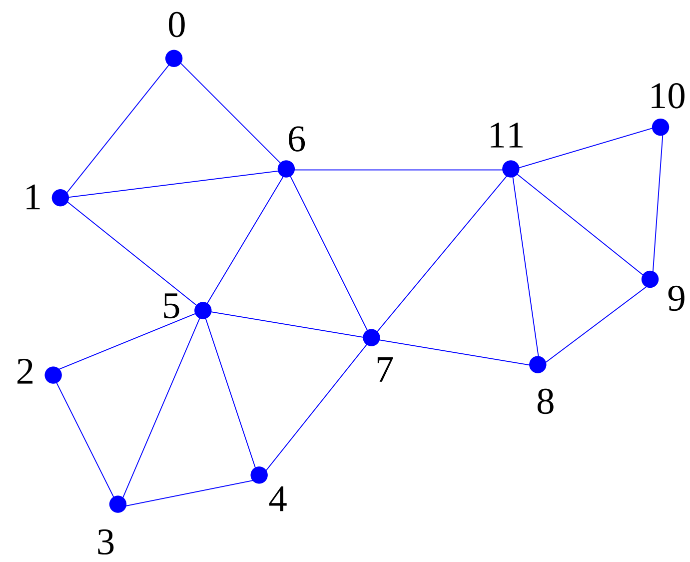
Graphical structure of triangulation is a 2-tree.

2-trees

A 2-tree is a graph defined by a set of "triangles" (3-cliques) connected along edges in a tree structure.

Every 2-tree admits a *perfect elimination order*:

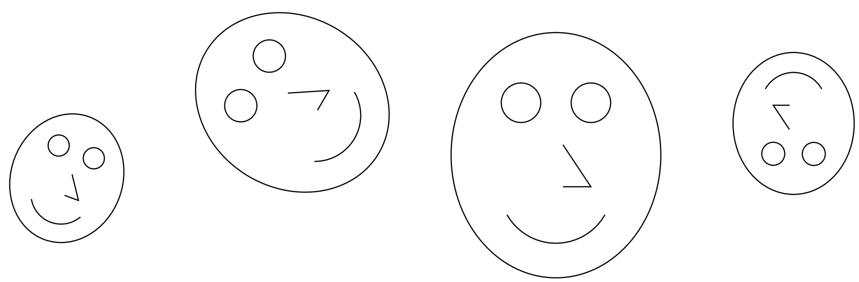
After eliminating the first i vertices, the next one is in a single triangle.



Two-dimensional shape

How does a triangulation help describe the shape of a polygon?

We say to objects have the same *shape* if they are related by a similarity transformation (translation, rotation, scale change).

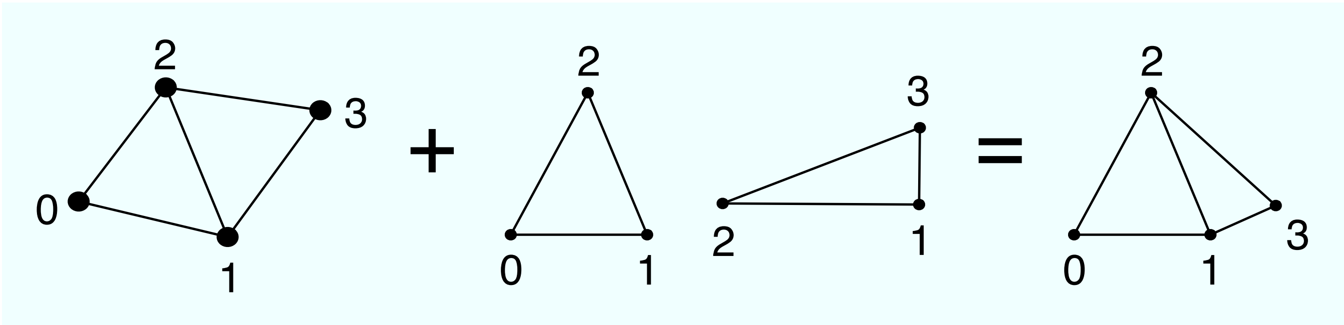


Different objects with the same shape.

Shape of triangulated polygons

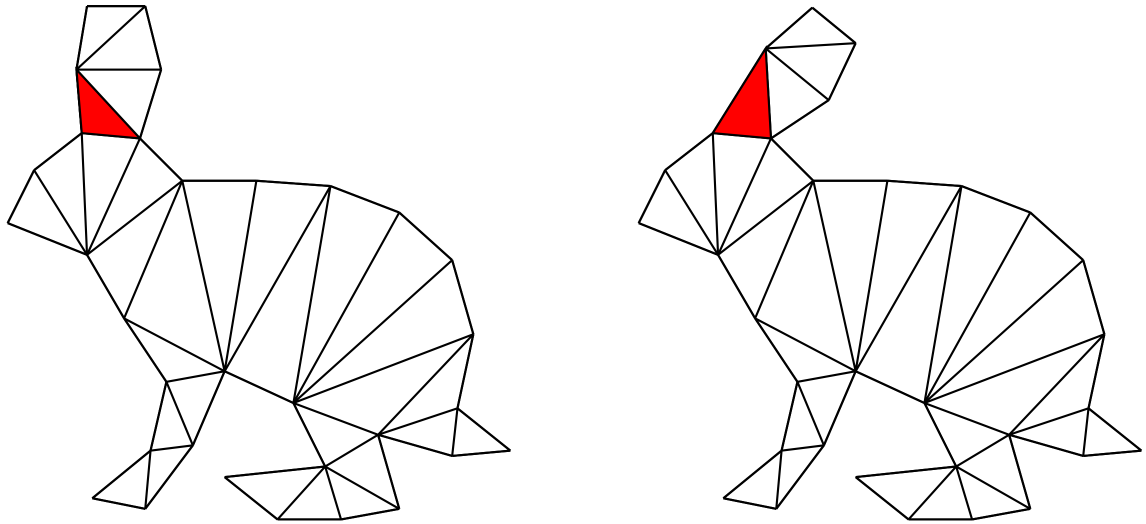
Say we have an object defined by the location of n vertices V, and $G = (V, E)$ is a 2-tree.

We can pick any shape for each "triangle" in G and obtain a unique shape for the object.



$\Rightarrow$ Shape of object is a point in $M_1 \times \cdots \times M_{n-2}$, where each M is a space of triangle shapes.

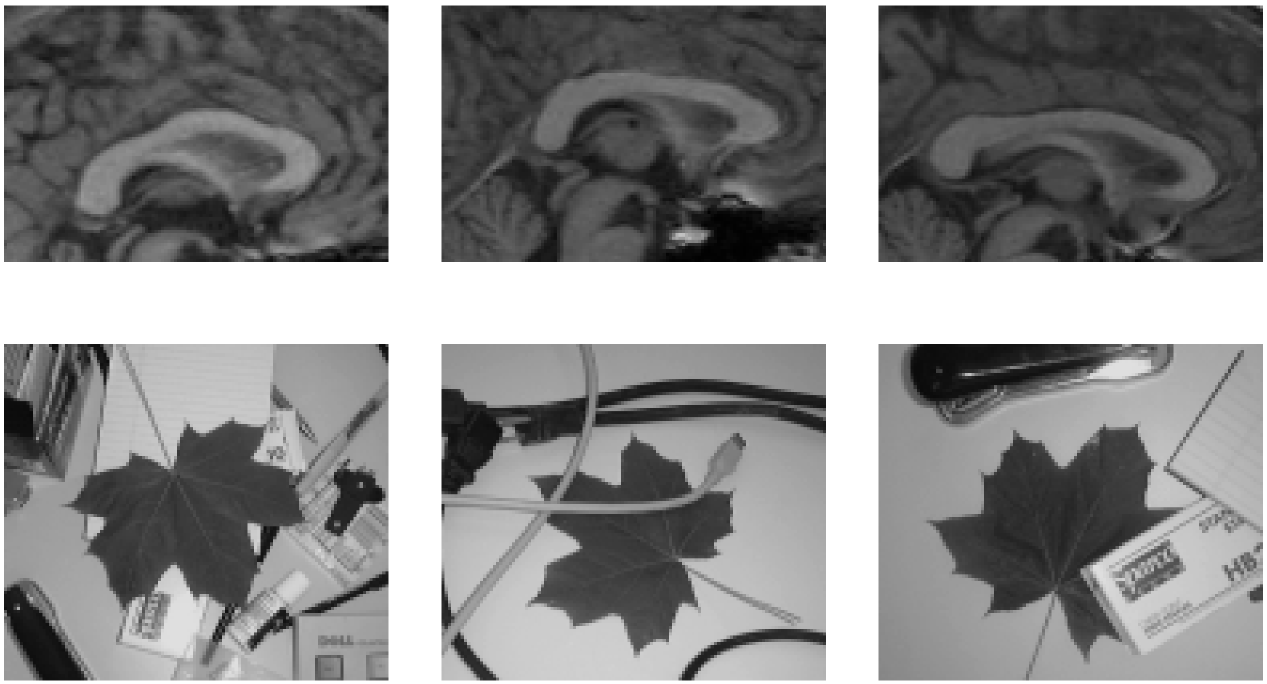
Deforming triangulated polygons



$$(x_1, \ldots, x_i, \ldots, x_{n-2}) \quad \rightarrow \quad (x_1, \ldots, x_i', \ldots, x_{n-2})$$

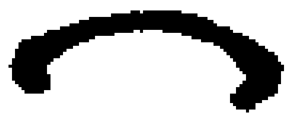
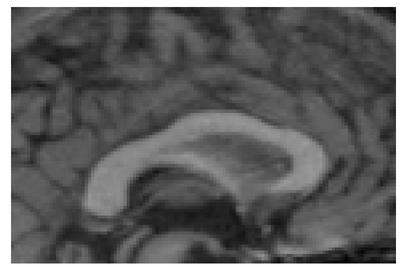
The rabbit ear can be bent by changing
the shape of a single triangle.

Finding non-rigid objects in images



Deformable template matching

Find “optimal” map from a template to the image.

$f :$  $\longrightarrow$ 

Quality of f depends on

- how much the template is deformed.
- correlation between the deformed template and image data.

Prior work

Most methods are based on local search techniques and depend on initialization near the right answer.

[Grenander et al.] Deformable boundary models.

[Widrow] Rubber masks.

[Cootes et al.] Active shape and appearance models.

Few methods are based on global optimization.

[Amit, Kong] Sparse landmarks.

[Coughlan et al.] Open curves.

Major challenges

- Represent both the boundary and the interior of objects.

- Capture natural shape deformations.

- Efficient matching algorithms:

  – Search for the *optimal* deformation - global minimum of cost function.

  – Initialization-free, invariant to rigid motions and scale.

Matching triangulated polygons

- Let T be a triangulation of a simple polygon P.

- Consider continuous maps $f : P \rightarrow \mathbb{R}^2$ that are affine when restricted to each triangle.
  - f takes triangles in the model to triangles in the image.
  - f is defined by where it sends the vertices of P.

- Quality of f is given by a sum of costs per triangle,

$$C(f, I) = \sum_{t \in T} C_t(f_t, I)$$

Example cost function

- Deformation cost for each triangle.
- Shape boundary is attracted to high gradient areas.

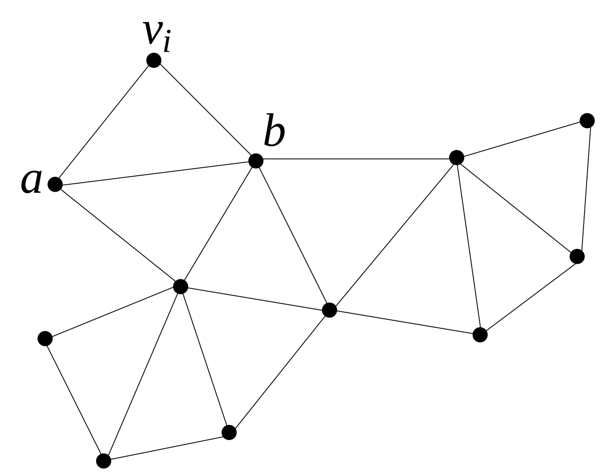
$$C(f, I) = \sum_{t \in T} \mathrm{def}(f_t) - \lambda \int_{\partial P} \frac{\|(\nabla I \circ f)(s) \times f'(s)\|}{\|f'(s)\|} \, ds$$

$\mathrm{def}(f_t)$ measures how far f_t is from a similarity transformation.

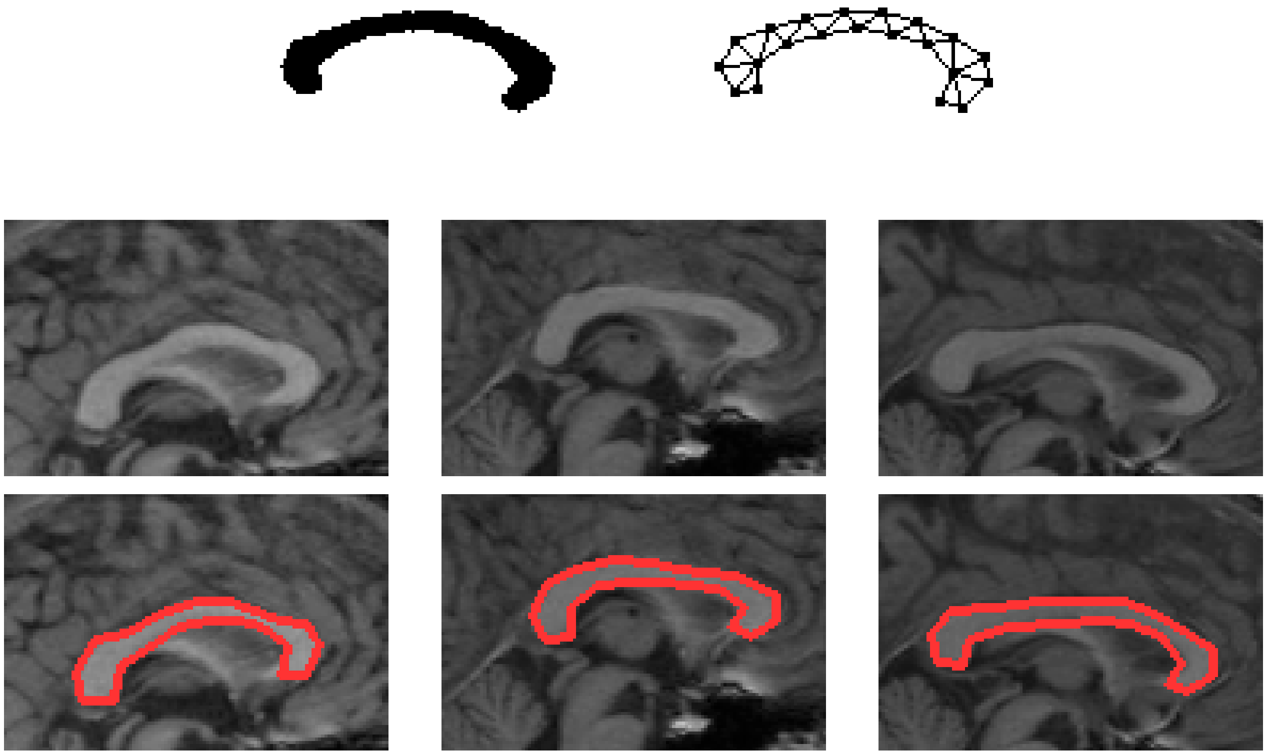
Combinatorial optimization

- Restrict $f(v_i)$ to be a location l_i in a grid $\mathcal{G}$.
- Dynamic programming algorithm using elimination order.
- Running time is $O(n|\mathcal{G}|^3)$, where n is number of vertices.

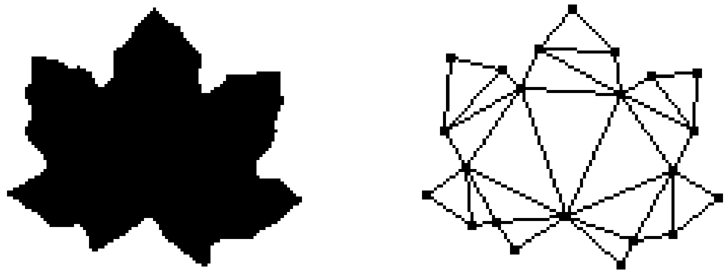
At step i, find optimal location for v_i as a function of locations of two other vertices.

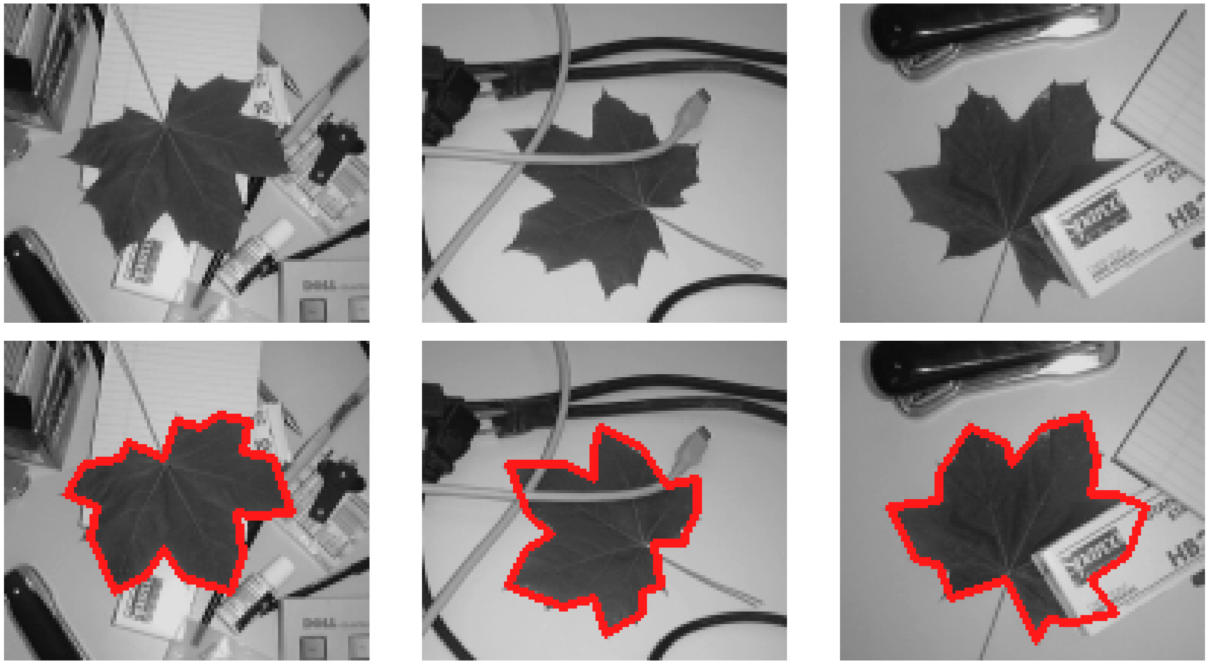


Matching results

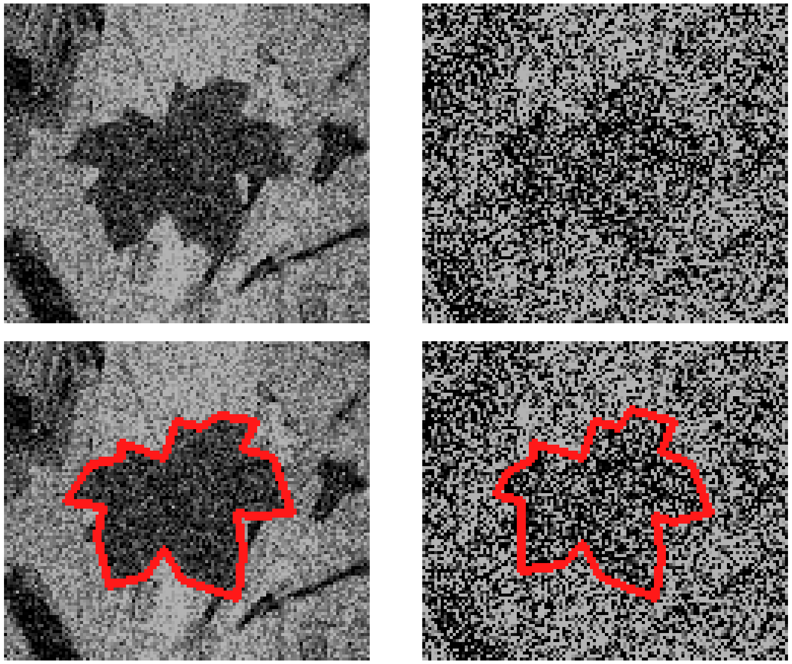


Matching results

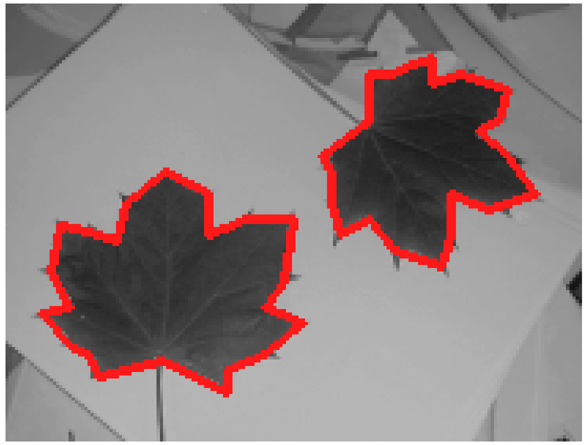




Matching results



Nosy images.

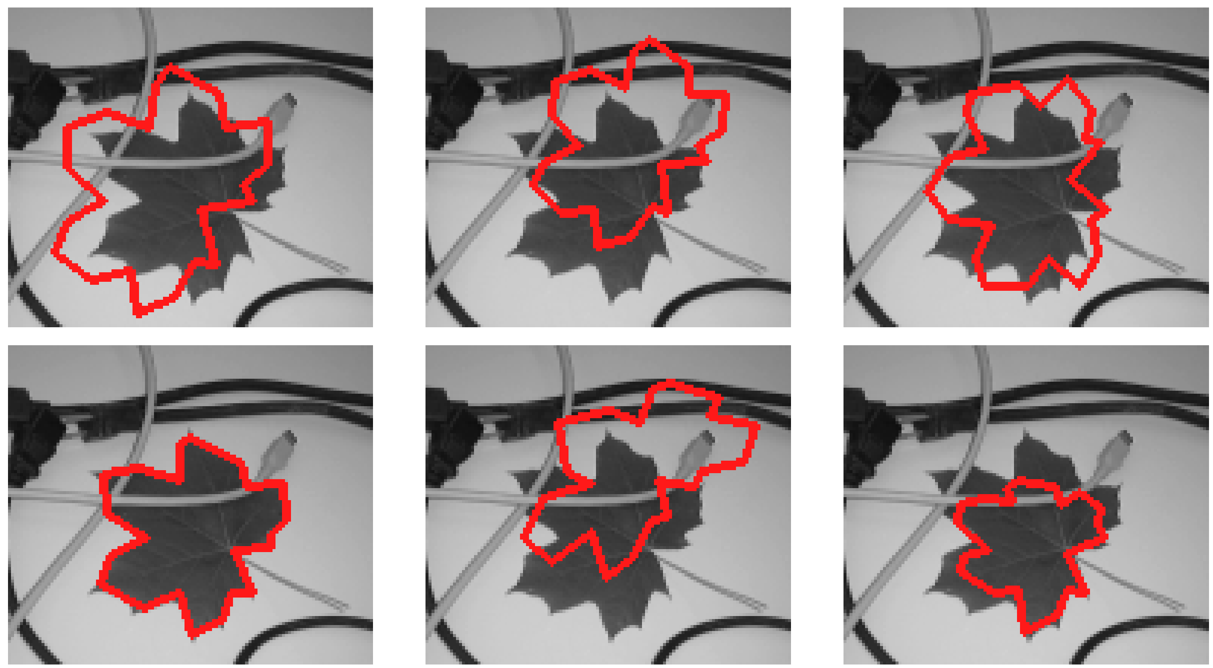


Multiple instances.

Contrast with local search method

Initialization:

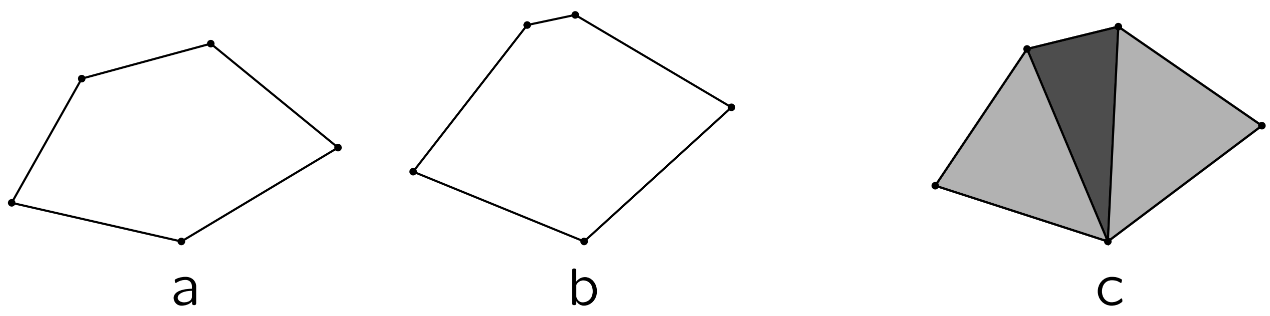
Result:



Learning models

Given multiple examples of an object,

- Pick a common triangulation.
- Learn shape model for each triangle (mean and variance).



Procrustes analysis

Given sample object configurations $\{X_1, \ldots, X_m\}$.

Assume each configuration comes from a mean by perturbation ϵ and similarity transformation g,

$$X_j = g_j(\mu + \epsilon_j)$$

Procrustes mean shape:

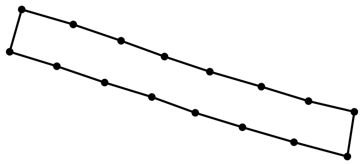
$$\widehat{\mu} = \underset{||\mu||=1}{\text{argmin}} \min_{g_j} \sum_{j=1}^{m} ||\mu - g_j X_j||^2$$

Learning triangulated models

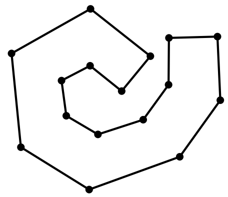
- Use Procrustes analysis for each possible triangle.
  - Local instead of global rigidity assumption.

- Select triangulation that can best represent examples.
  - There is a cost associated with each triangle. (corresponding to how rigid it is)
  - Can use dynamic programming to select optimal one.

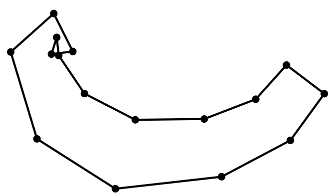
Local versus global rigidity assumption



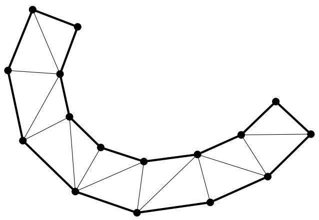
a



b

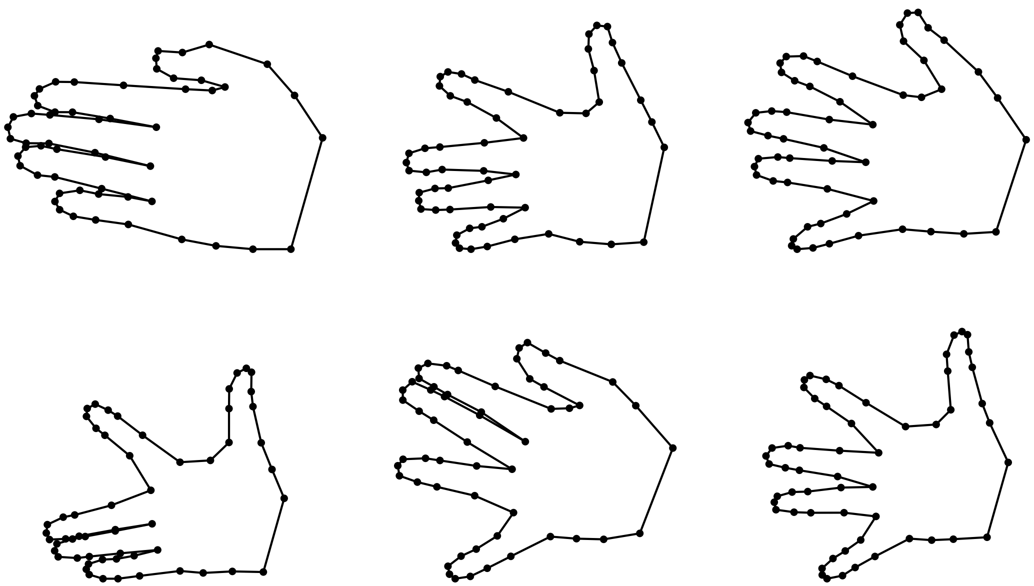


Procrustes mean



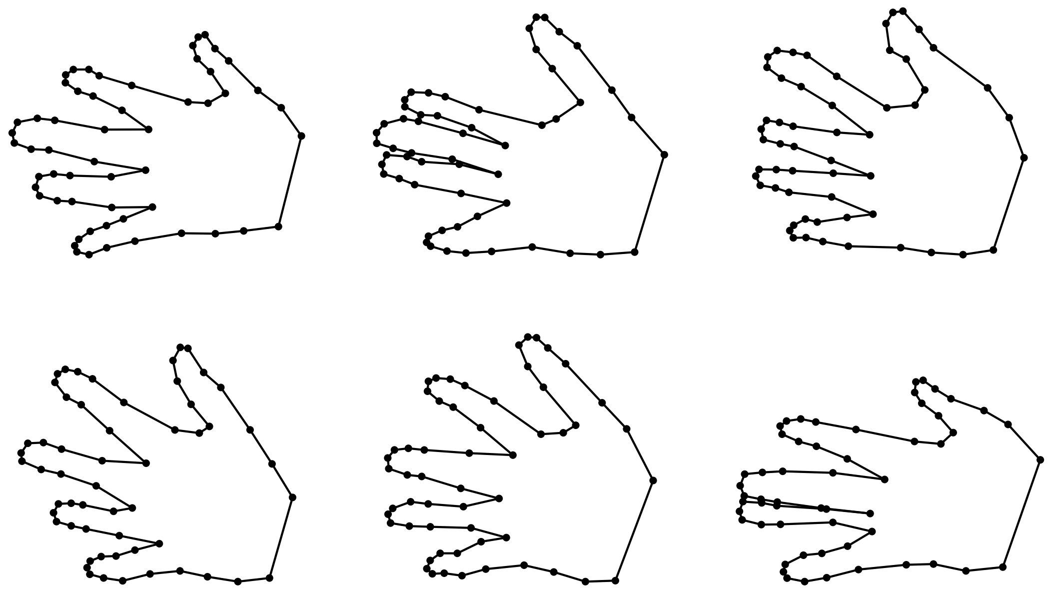
Triangulated model

Hands



A few of a total of 40 samples of hands from multiple people.

Typical deformations of learned model

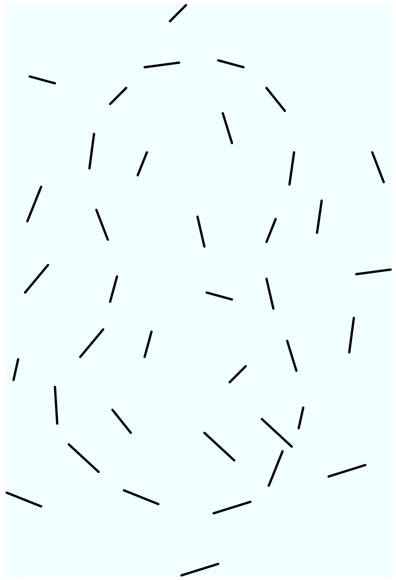


Random samples from the prior model for hands.

Shape grammar

Finding objects in images without using specific models.

- Build a generic shape model to capture properties of "typical" objects.
- Gestalt laws: continuity, smoothness, closure, symmetry, etc.



Shape grammar

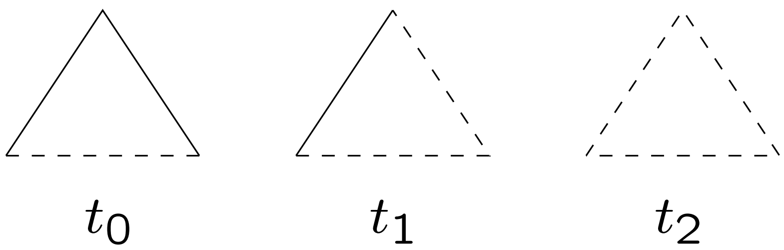
- Define a stochastic growth process that generates triangulated polygons using a context free grammar.

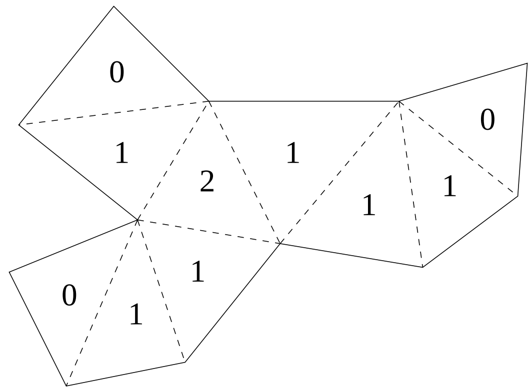
- The grammar can generate any triangulated polygon, but it tends to generate shapes with certain properties.

  - Gives a generic model for objects.

  - Captures which are good interpretations of a scene.

Shape tokens





- t_0 corresponds to ends of branches.
- sequences of t_1 correspond to branches.
- t_2 connects multiple branches together.

Growing a shape

- A root triangle of type i is selected with probability p_i.

- Each dotted edge “grows” into a new triangle.

- Repeat until there are no more dotted edges.

- For each triangle type there is a distribution over its shape.

Structure

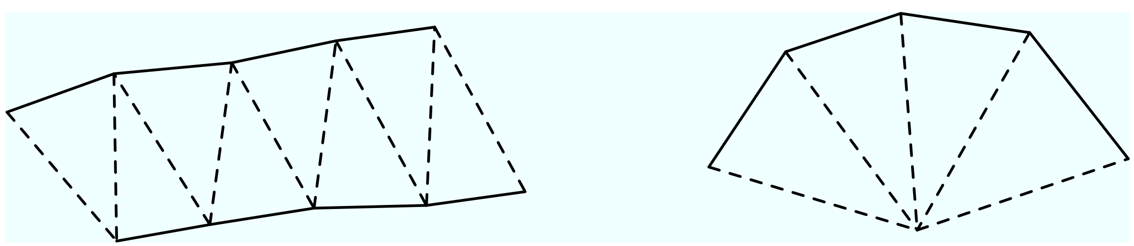
- Growth process always terminates when $p_2 < p_0$.
- Expected number of triangles is $E[n] = 2/(p_0 - p_2)$.
- Expected number of branching points is $E[j] = 2p_2/(p_0 - p_2)$.
- Together $E[n]$ and $E[j]$ define p_0, p_1 and p_2.
- Typically $p_1 \gg p_0, p_2$.

Geometry

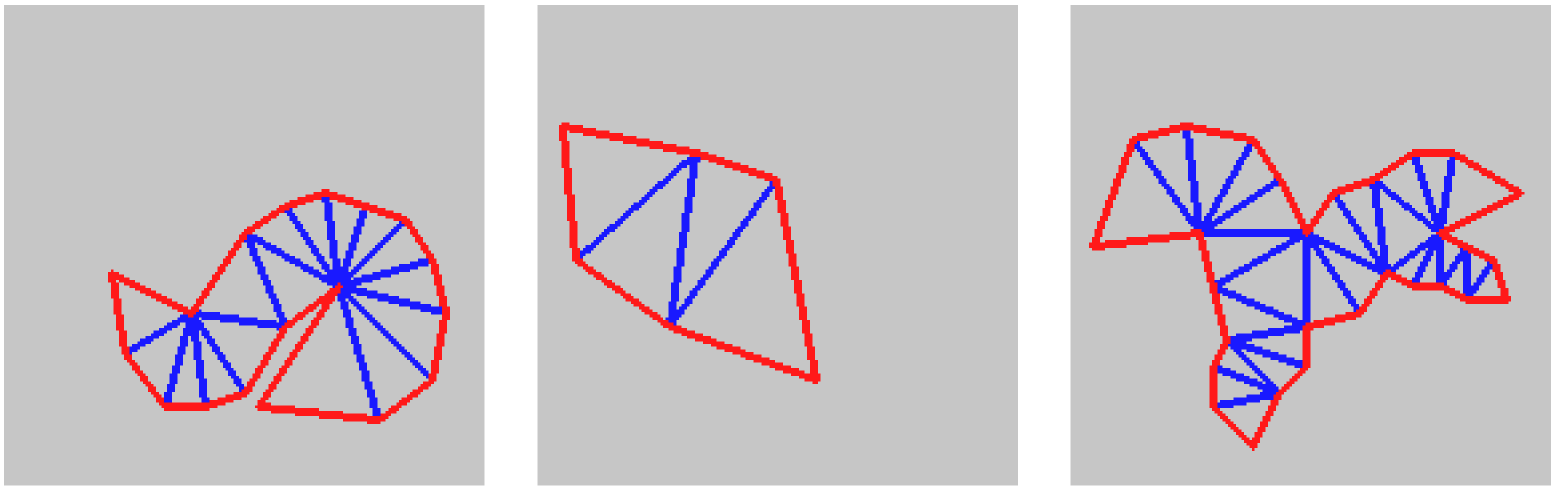
If t_1 is skinny and isosceles,

- shapes have smooth boundaries almost everywhere.

- each branch tends to have axial symmetry.



Random shapes



Finding objects in images

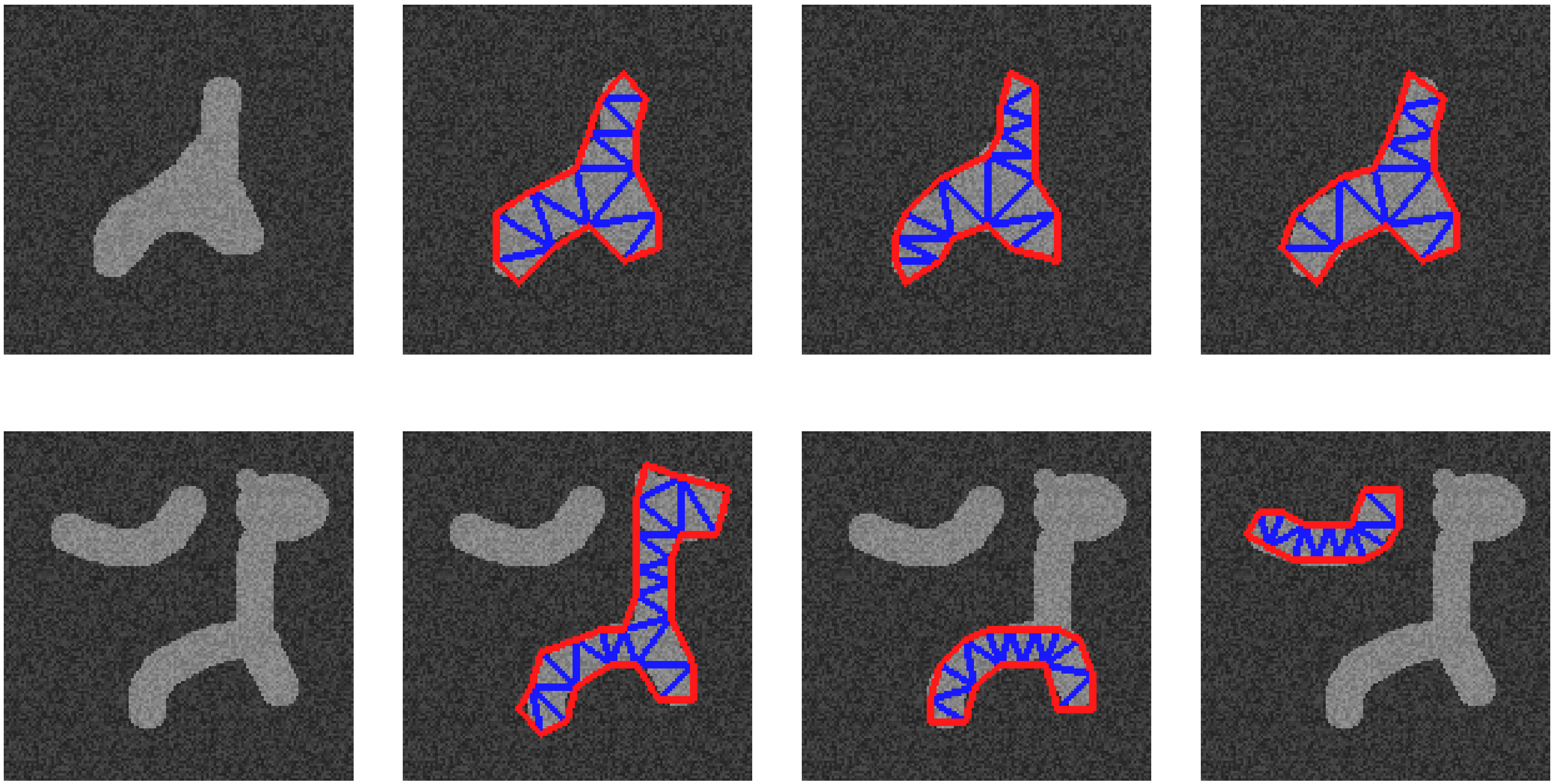
- Grammar generates random objects, without taking into account image data.

- Look for triangulated polygons that align with image features and would likely be generated by the grammar.

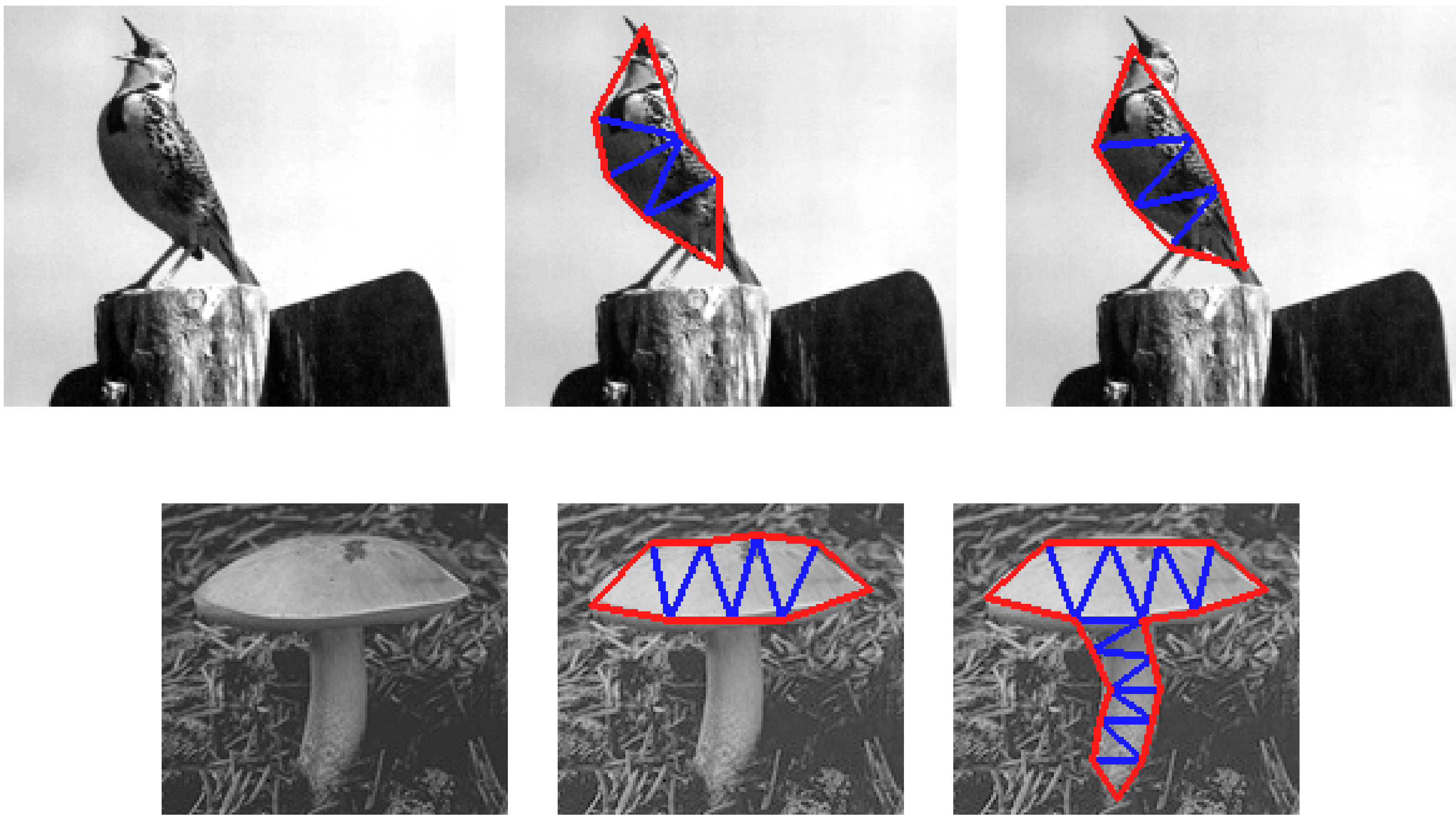
  – These are good hypotheses for objects in the scene.

- This process generates possible interpretations of a scene and a separate process can verify each one.

Example results



Example results



Summary

- Representation of objects using triangulated polygons.
  - Detecting deformable objects.
  - Learning deformable shape models from examples.
  - Detecting generic objects.
- Future work:
  - Richer grammars, with intermediate structure.
  - Shape classification.