Part-Based Recognition

Benedict Brown

CS597D, Fall 2003

Princeton University

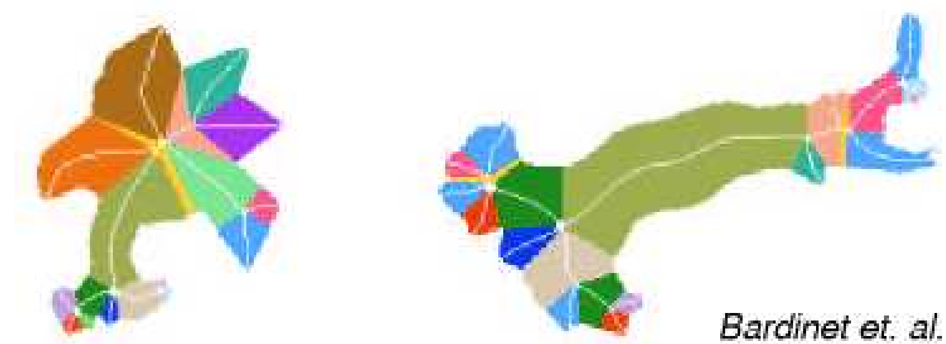
Introduction

- Many objects are made up of parts
- It's presumably easier to identify simple primitives than complex shapes
- Object can be characterized by relationship between primitives

- Some research suggests humans identify objects this way
- Works in both 2D images and 3D data sets

Two Approaches

- Template Matching: Fit representation of parts to shape templates in order to classify object
- Direct Matching: Compare part representations of two objects in order to evaulate similarity

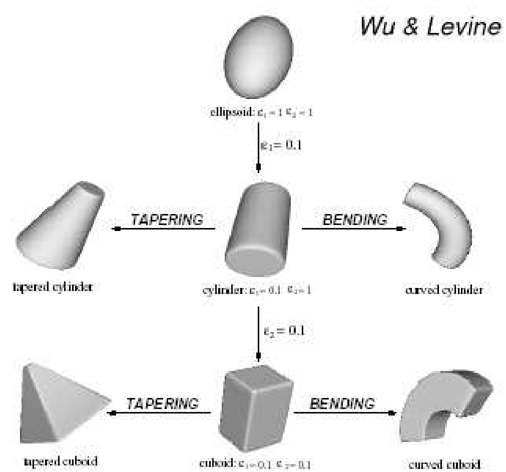


Overview

- Representations
  - Geons: Identify simple primitives with properties which are invariant to viewpoint, and encode their spatial relationships (Biederman, Wue & Levine)
  - Parametric models (generalized cylinders, superquadrics, *etc.*), and spatial relationships of these (Wu & Levine, Min)
  - Segmented models (Anderson, *et. al.*)
- Template Matching

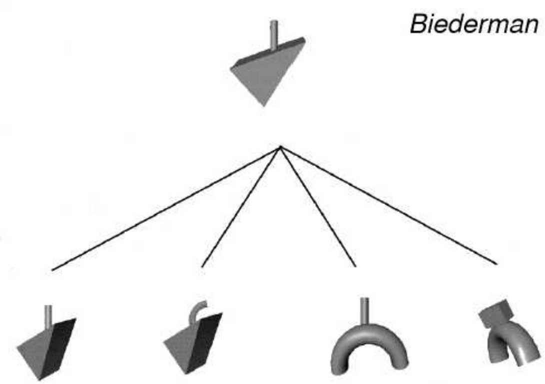
Geons

- Geons are simple parts having viewpoint-independent *Non-Accidental Properties* (NAPs)
- Relationships between parts (*e.g.* “perpendicular to”) are used to encode shape in a *Geon Structural Description* (GSD) graph
- First proposed by Biederman

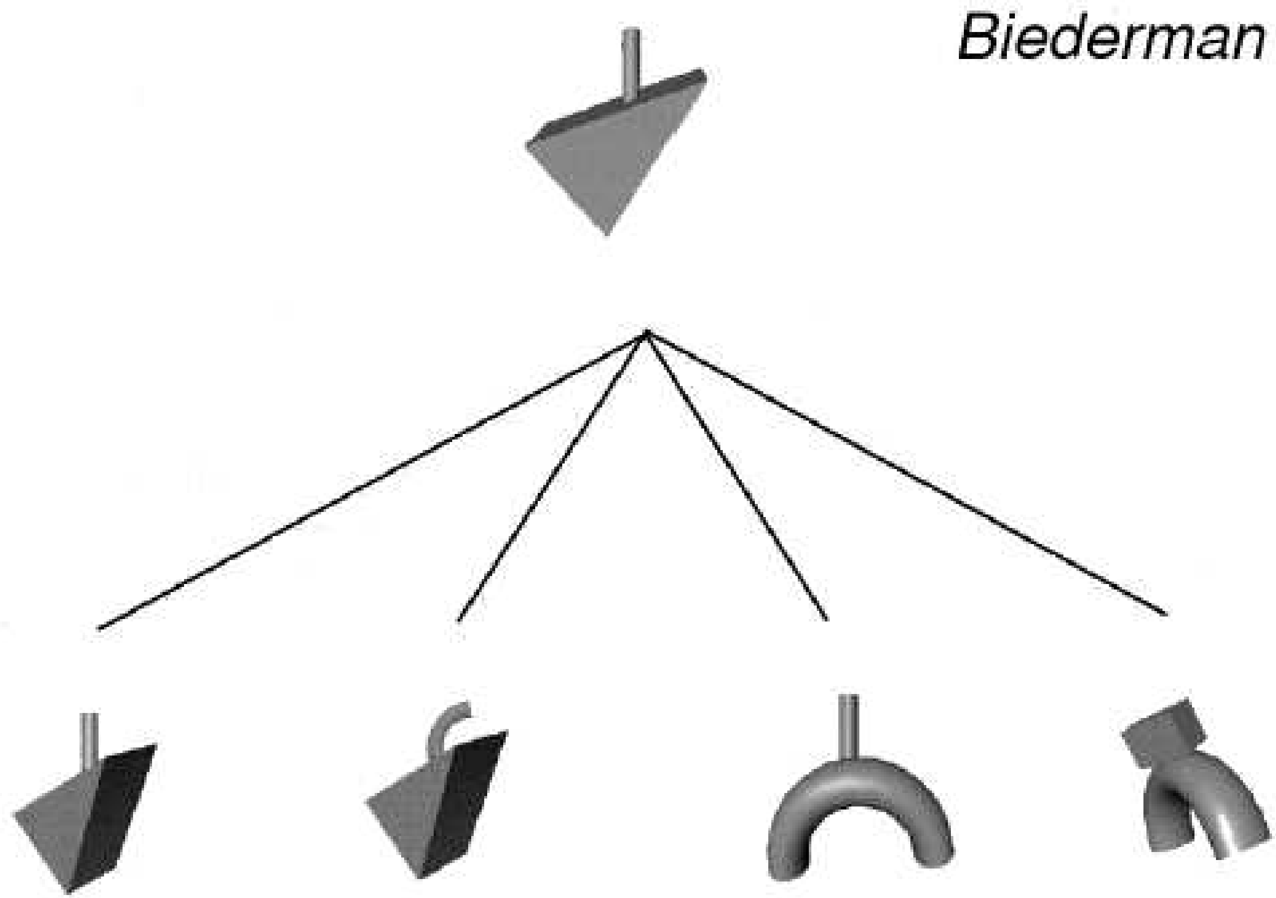


Non-Accidental Properties

- Viewpoint-invariant properties of a geon
- Can be used to reconstruct of a geon from almost any 2-D view
- For example, curved vs. straight edge, parallel, symmetric, tapered, *etc.*
- In practice, not all NAPs of a geon will be visible in every viewpoint



Geon Motivation



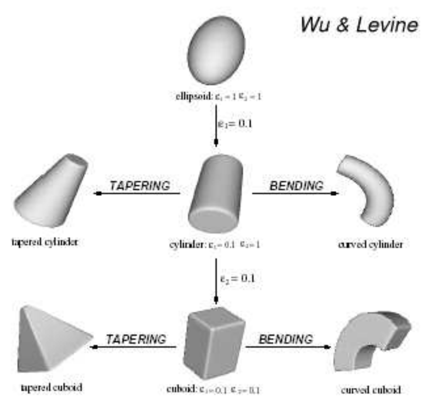
Implementing Geons in Images

- Extract edges and contours
  - Biederman's implementation only works on stylized line drawings
  - Dickinson, *et. al.* use user-segmented images
- Extract NAPs
- Match to geons using a neural net
- Construct GSD from geons, and match to objects in databse

Implementing Geons in Range Data

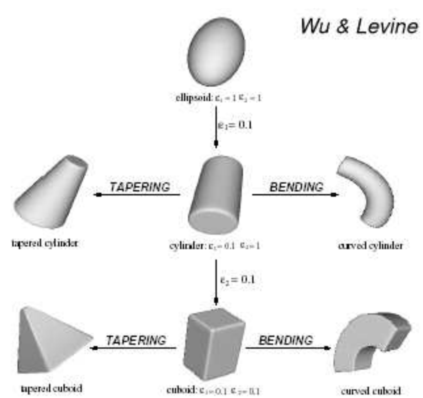
- Segment range data (Wu & Levine assume single-part data)
- Restrict to only seven different geons, parametrized as subset of superellipsoids:

$$\left(\left| \frac{x}{a_1} \right|^{2/\epsilon_2} + \left| \frac{y}{a_2} \right|^{2/\epsilon_2} \right)^{\epsilon_1/\epsilon_2} + \left| \frac{z}{a_3} \right|^{2/\epsilon_1} = 1$$



Implementing Geons in Range Data

- Different values of ϵ_1 and ϵ_2 yield ellipsoid, cylinder and cuboid.
- Additionally apply linear tapering or circular bending to each primitive
- Search for best match to data in terms of both surface and normal matching



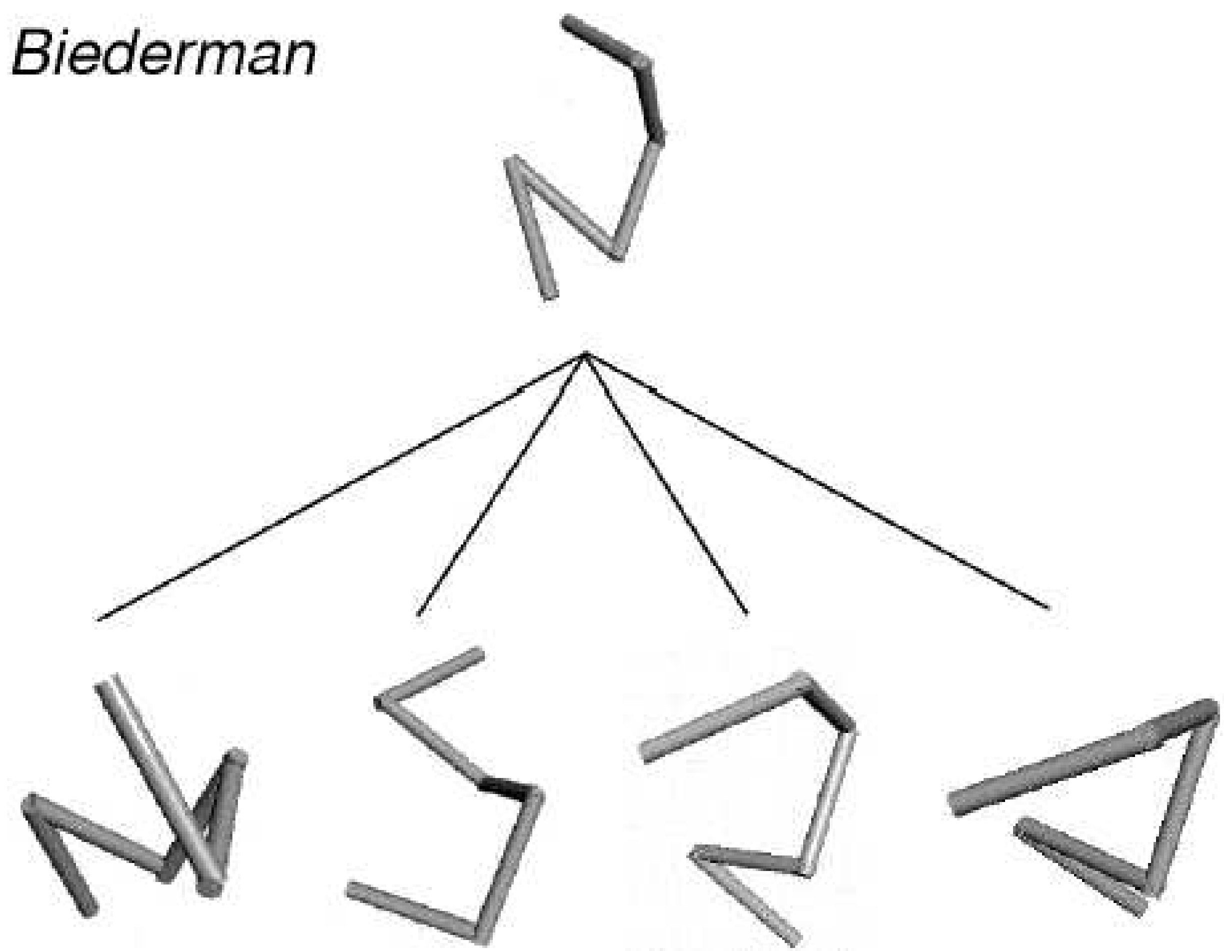
Advantages of Geons

- Research indicates humans may use this kind of recognition for objects which decompose easily into parts
- Many common objects (*e.g.* tea kettles) decompose easily into parts
- Simple, expressive idea
- Avoids intractable training problem by limiting the number of primitive objects a computer must be able to recognize

Disadvantages of Geons

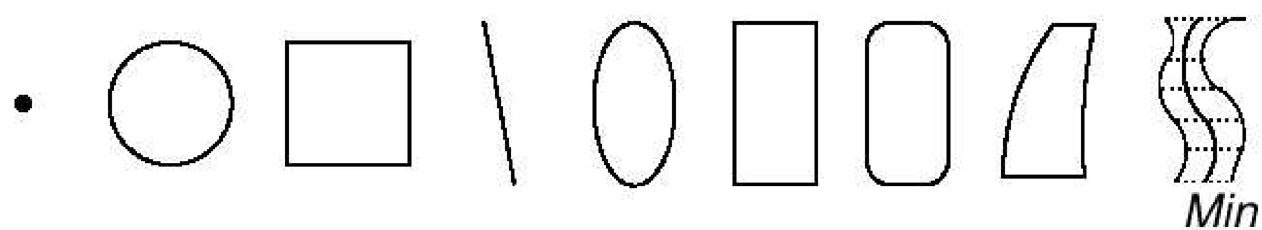
- Research indicates humans may not use this kind of recognition for objects which decompose easily into parts
- Many common objects (*e.g.* trees) do not compose easily into parts
- Current vision algorithms are totally incapable of robustly performing the necessary segmentation and contour detection

Non Part-Based Object



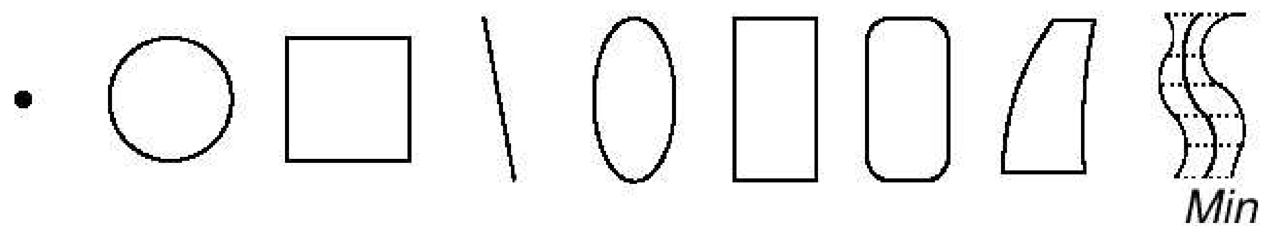
Parametric Models

- Fit a collection of mathematical constructs to a shape
- In 2-D, arrange constructs so contours conform to image contours (Dickinson, *et. al.*)



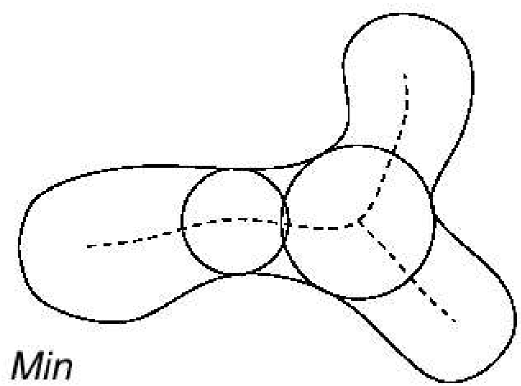
Parametric Models

- Unlike geons, these encode actual shape, not NAPs
- Possible constructs: generalized cylinders, superquadrics, algebraic surfaces
- In practice, recovering geons often involves fitting parametric models, then discarding parameters (Wu & Levine)



Segmented Models

- Classical algorithms: Medial axis transform, skeleton, color-based segmentation
  - Especially on 2-D models, results are insufficient
- Can be couple with template-based freeform deformation models to obtain good segmentation information (Lieu & Sclaroff)

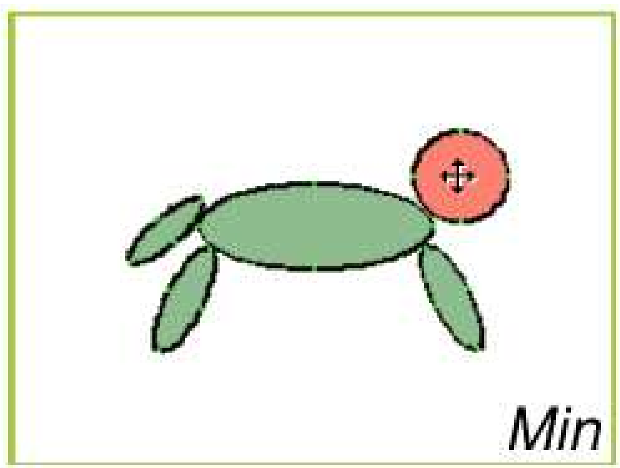


Overview

- Representations
- Template Matching
  - Encode and match structural relationships in 2-D view (Min)
  - Freeform deformation: Warp an *a priori* 3-D template to fit it to a portion of the data (Anderson *et. al.*)

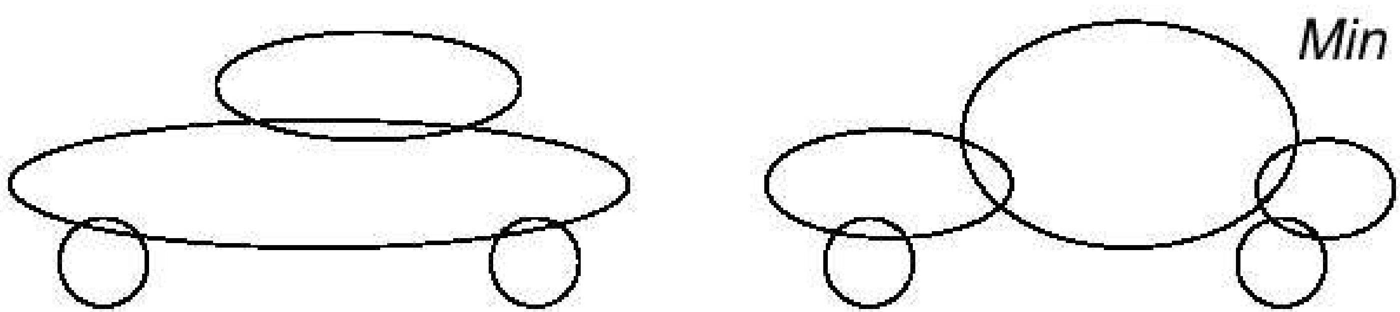
2-D Shape Query Interface

- Goal: Find objects in a model database, base on object structure
- Design Issues: Must be fast, with simple UI
- Solution: User represents structure with ellipses, computer generates template and matches to 2-D views of models



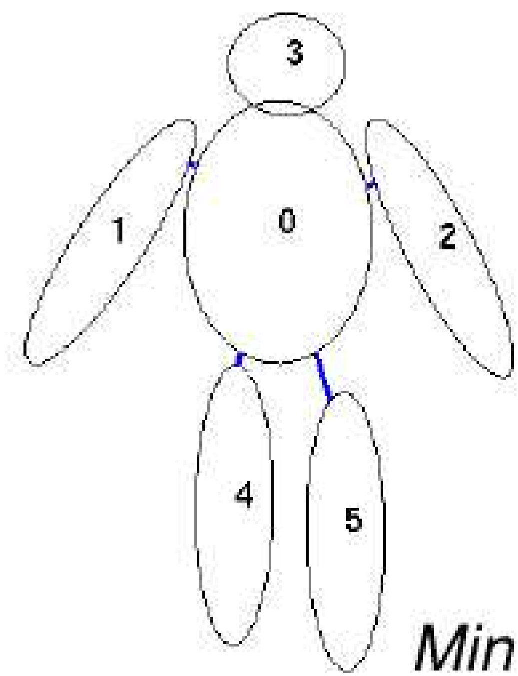
Shape Decomposition

- Ellipses are computationally simple, ease to draw, and expressive
- Ellipses do *not* provide a unique structural description



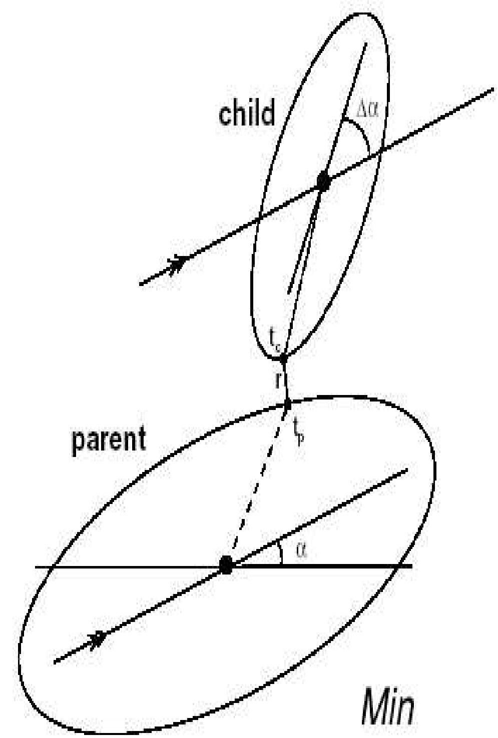
Matching Approach

- Construct graph of ellipses
- Match to 2-D views of database models, while trying to minimize ellipse deformations



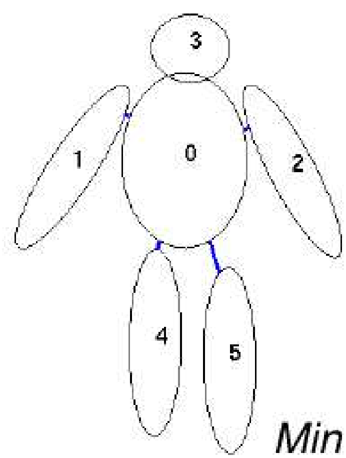
Graph Structure

- User selects "root" ellipse
- All ellipses connected indirectly to root via tree
- Child ellipses are parametrized in terms of:
  - t_p: attachment point on parent
  - t_c: attachment point on child
  - r: distancee between attachment points
  - $\Delta\alpha$: relative angle of the child
  - relative scale, aspect ratio



Tree Construction

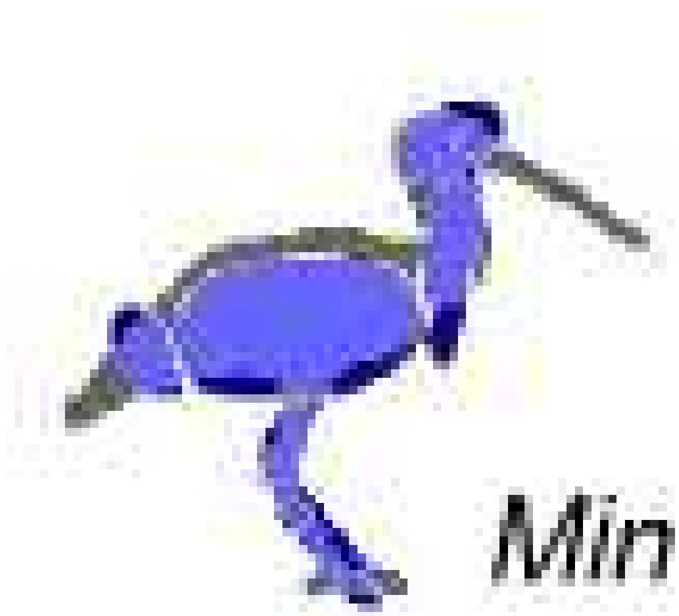
- Root ellipse is selected by user
- Construct weighted graph between of all pairs, with edges weights of $wd^2 + 1/\sqrt(a_1 + a_2)$, where w is a weighting term, d is the shortest distance between the two ellipses and a_1 and a_2 are their areas
- Construct MST



Objective Matching Function

Image overlap: How well does the template cover the model?

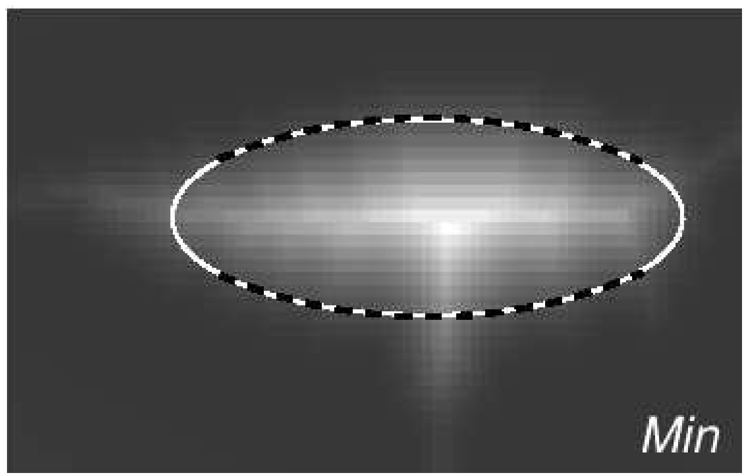
$1 - n_c/n_i$ where n_c is the number of model pixels covered by ellipses, and n_i is the total number of model pixels in image



Objective Matching Function

Part Alignment: How well do ellipses align with model?
Compare ellipse to Euclidean Distance Transform. Values along major axis should be equal to major axis length, and values at boundary should be zero. Obtain robustness by through sampling and averaging.

Term averaged over all ellipses



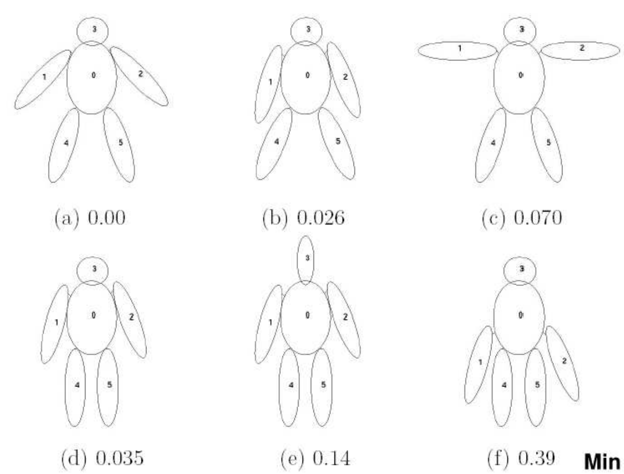
Objective Matching Function

Parts Deformation: How much has the template deformed?

For root ellipse: $w(1 - ar_1/ar_2)^2$, where w is a weight, and ar_1 and ar_2 are the original and new aspect ratios

For all other ellipses, use weighted sum of change in their parameters relative to parent

Term averaged over all ellipses

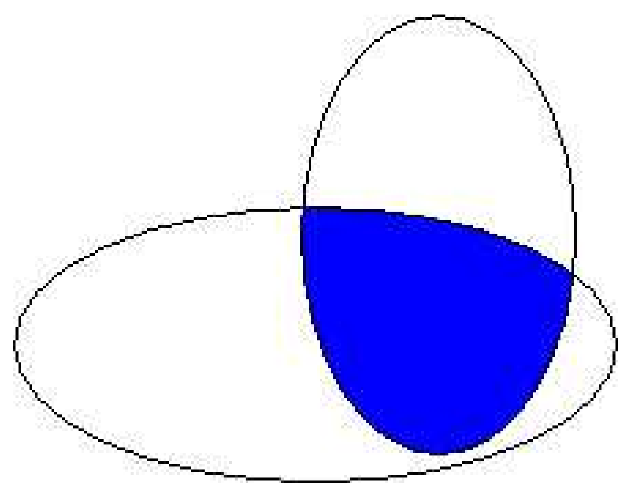


Objective Matching Function

Parts Overlap: How much do ellipses encroach on each other?

Sample k points on each ellipse. For each pair of ellipses, error is $(n_i - c_i)/k$ where n_i is number of points which fall in the other ellipse, and c_i is the number of overlaps in the original template

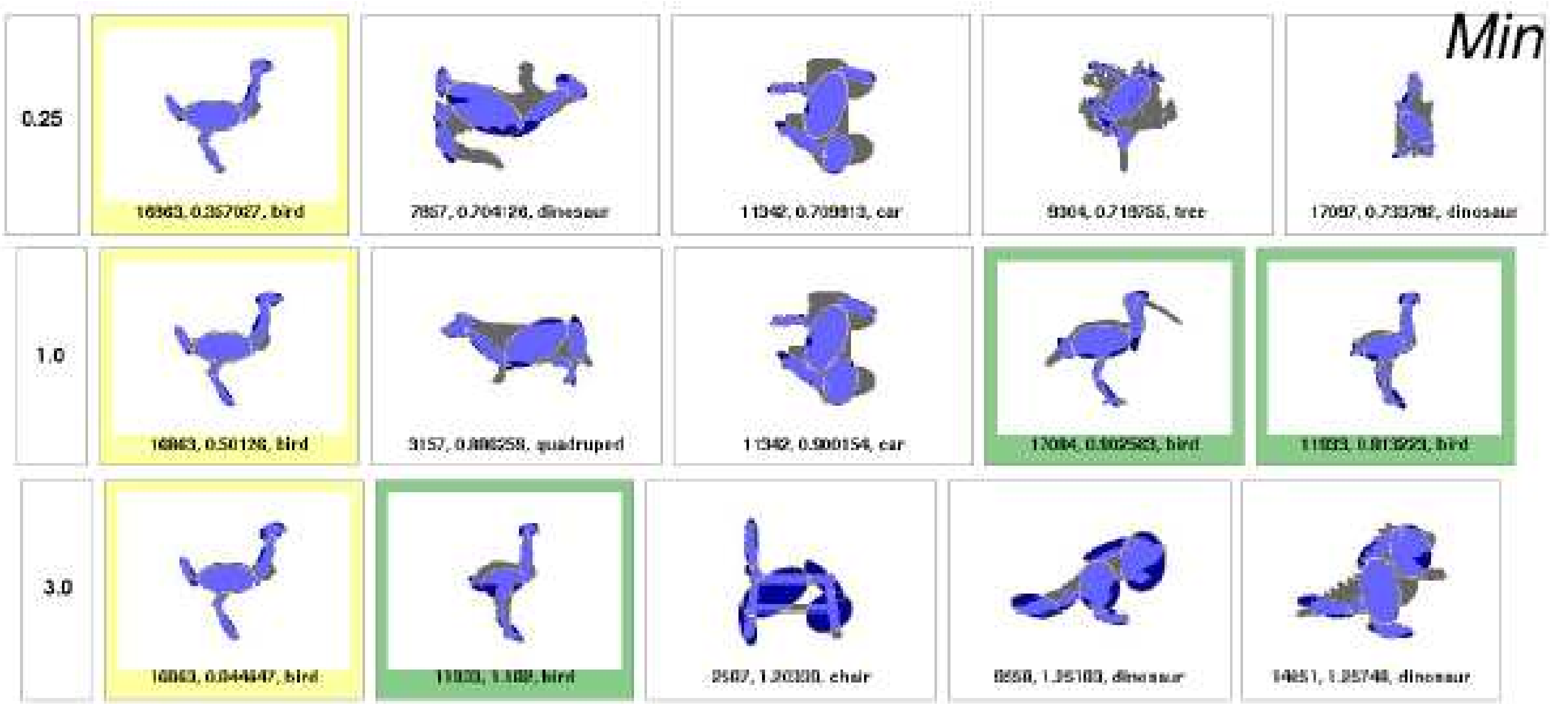
Term averaged over all pairs of ellipses



Minimization Method

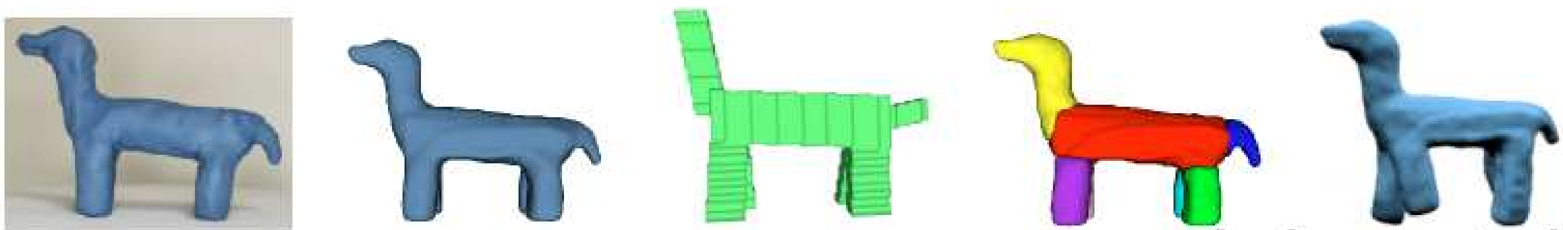
- Normalize translation using center of mass
- Normalize scale using average radius
- Recover rotation by trying dense sampling of rotations
- Optimize using the multidimensional downhill simplex method

Results



Volumetric Templates

- Goal: Animate digitally scanned clay models
- Approach: Fit volumetric deformable templates to scans to identify object, then animation is easy

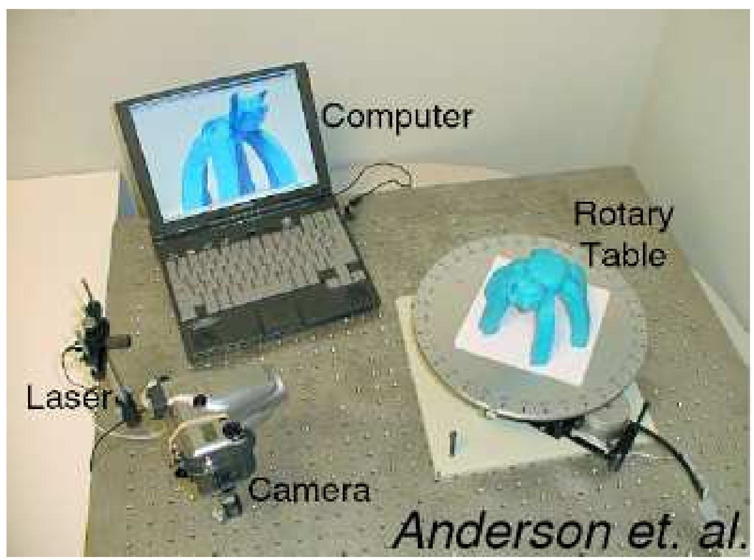


Anderson et. al.

Scanning Method

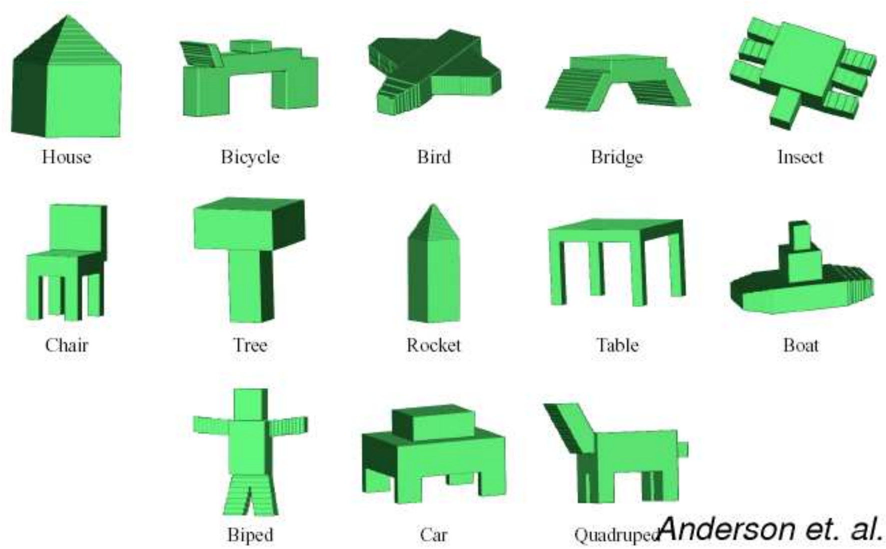
Turntable scanner, with object placed in canonical starting orientation

- Canonical starting orientation eliminates need for rotation invariance, and allows robust size and translation scaling via bounding box



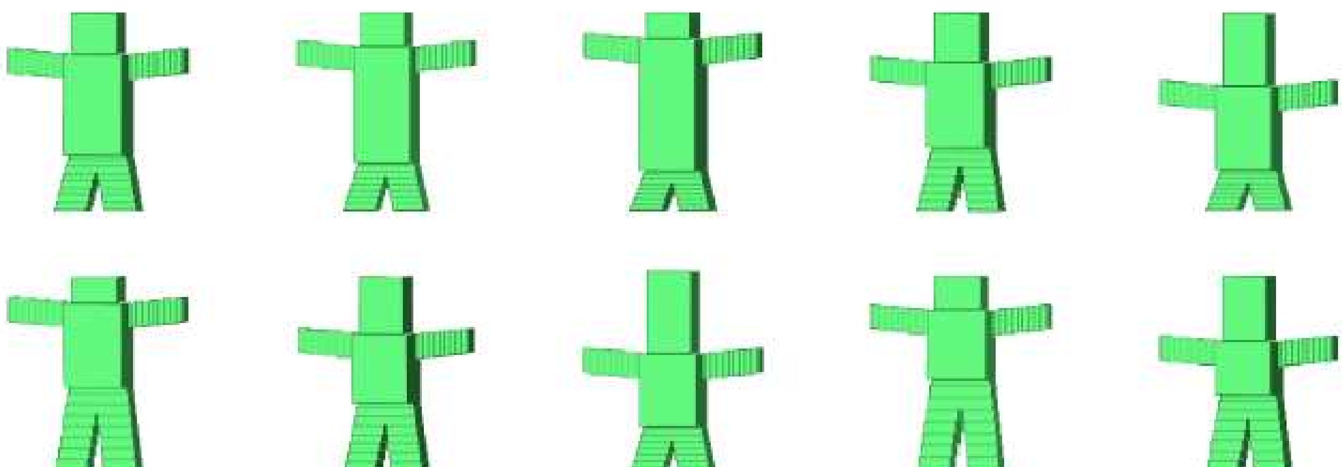
Template Design

- Templates are articulated, truncated rectangule pyramids (beams)
- User Created, 10 per object class with varying beam parameters
- 6 parameters per beam implies 60 parameters for biped model, but iter-beam constraints reduces this to 25



Template Design

- Templates are articulated, truncated rectangule pyramids (beams)
- User Created, 10 per object class with varying beam parameters
- 6 parameters per beam implies 60 parameters for biped model, but iter-beam constraints reduces this to 25

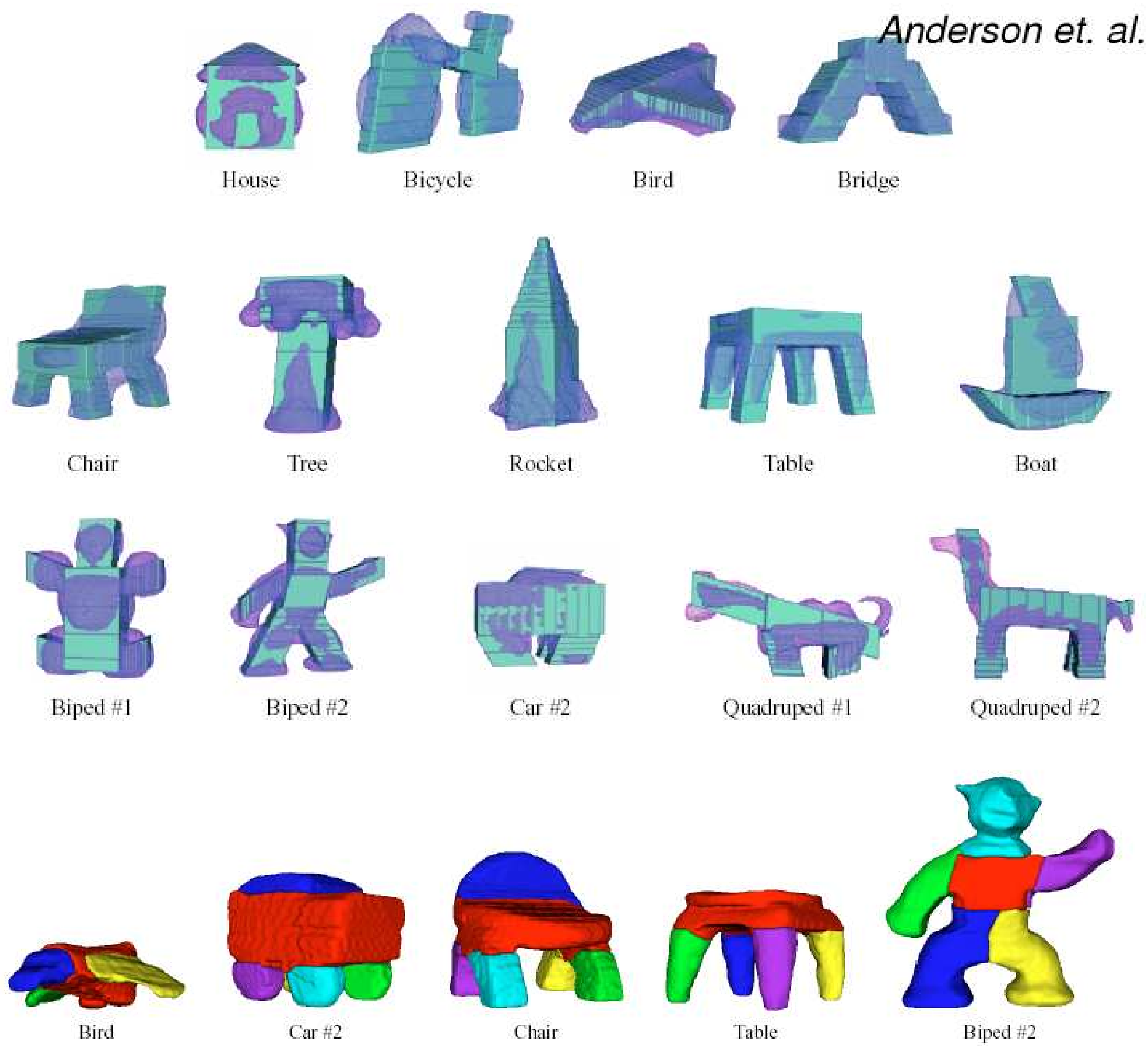


Anderson et. al.

Matching Algorithm

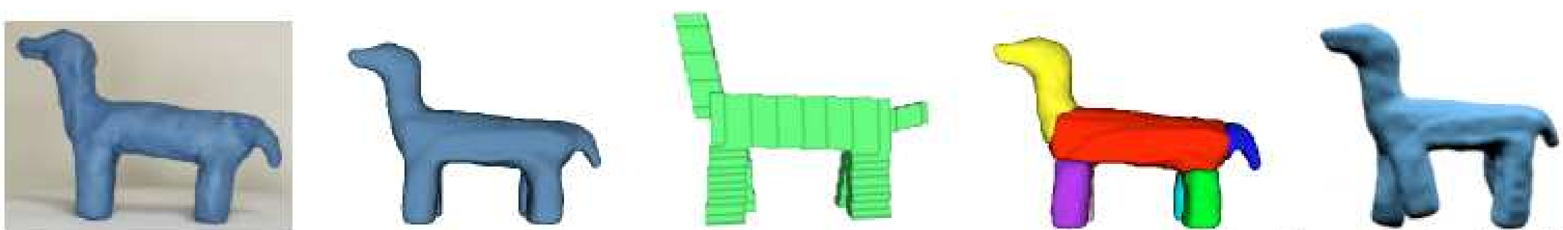
- Objective Function
  - Rasterize model to $128 \times 128 \times 128 voxels$
  - $+1$ for each superposition of model and template voxel
  - $-r$ for each extra template voxel, where r is the distance to the closest model voxel
  - Exponential deformation penalty based on distance between original and current template parameter vectors
- Gradient descent minimization

Classification Results



Animating Results

- Template are articulated so they can be animated
- Given matching of clay model to a template, it can be animated too
- Each beam influences every model voxel in inverse proportion to square of its distance from voxel
  - This helps prevent tears and cracks



Anderson et. al.