CSE373: Data Structures & Algorithms

Lecture 23: Applications

Linda Shapiro

Winter 2015

Announcements

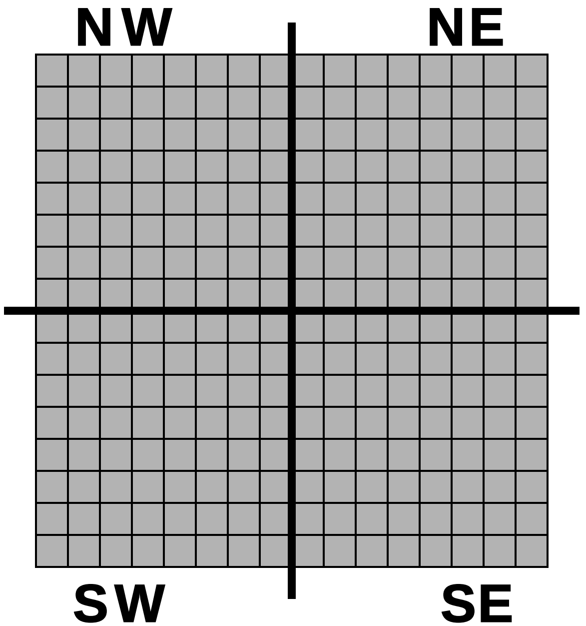
Other Data Structures and Algorithms

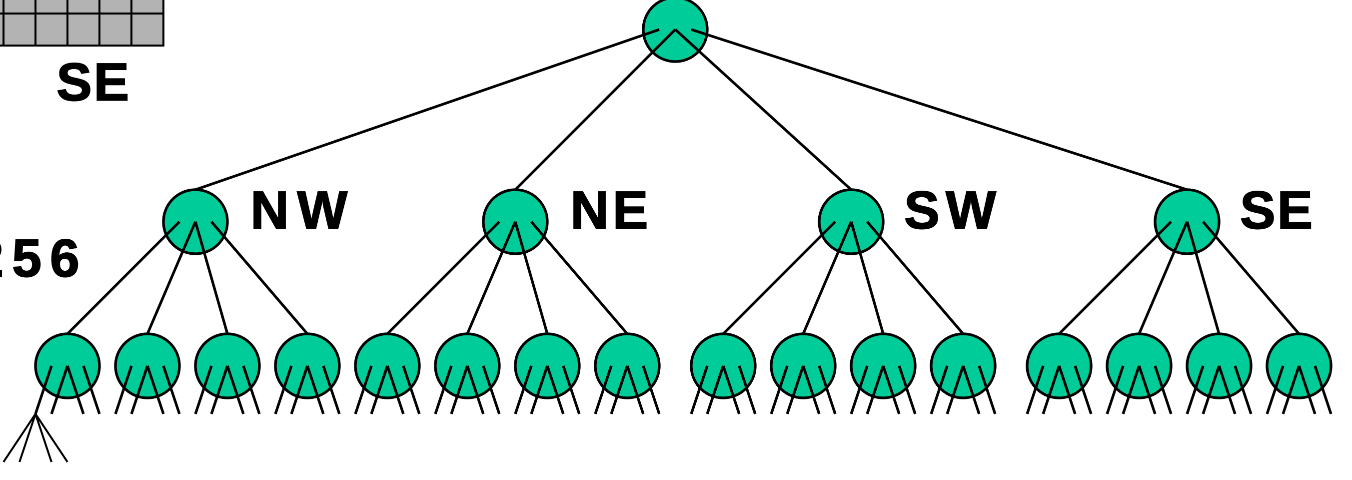
- Quadtrees: used in spatial applications like geography and image processing
- Octrees: used in vision and graphics
- Image pyramids: used in image processing and computer vision
- Backtracking search: used in AI and vision
- Graph matching: used in AI and vision

Quadtrees

- Finkel and Bentley, 1974
- Lots of work by Hanan Samet, including a book
- Raster structure: divides space, not objects
- Form of *block coding:* compact storage of a large 2-dimensional array
- Vector versions exist too

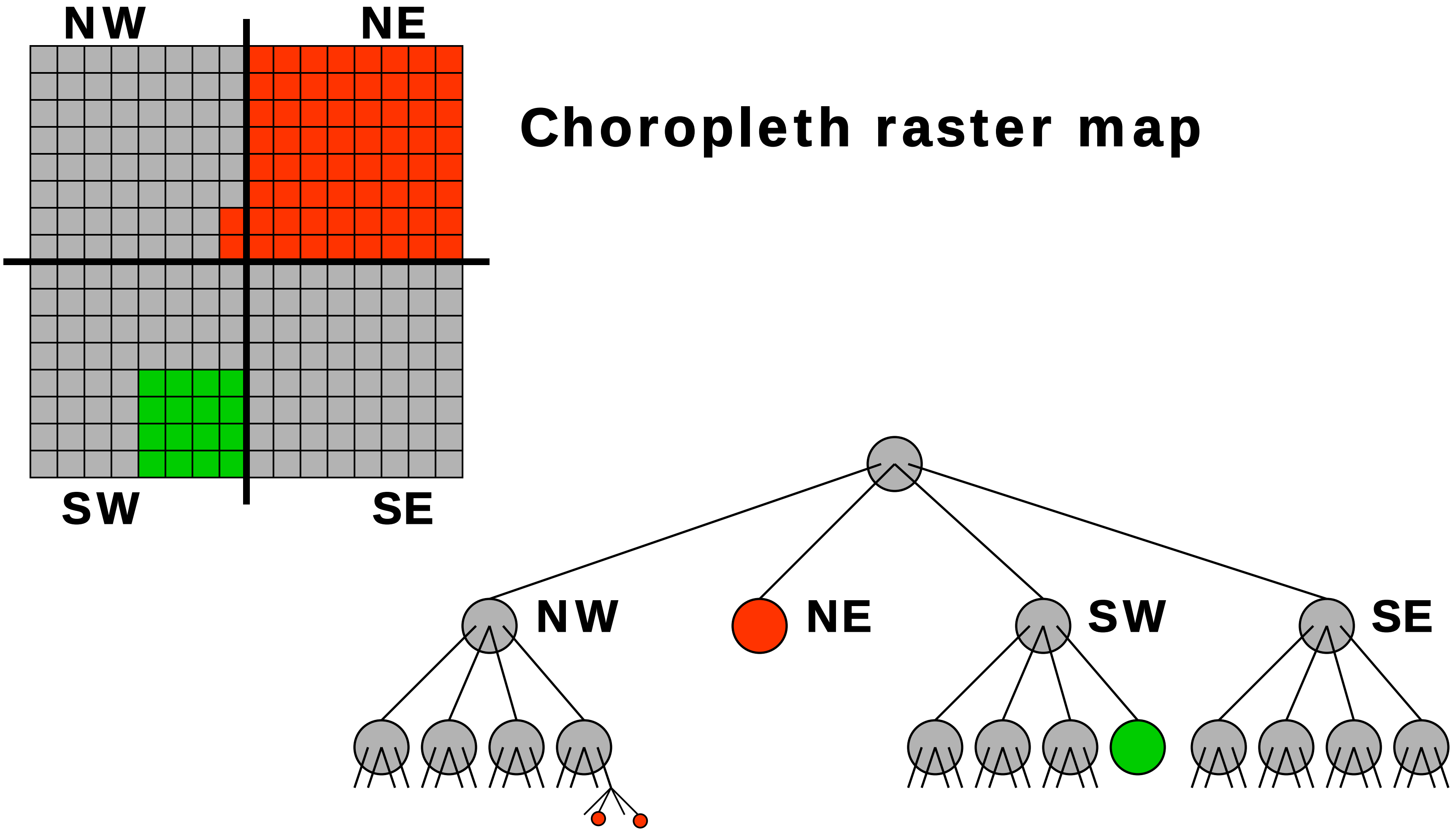
Quadtrees, the idea





1, 4, 16, 64, 256 nodes

Quadtrees, the idea



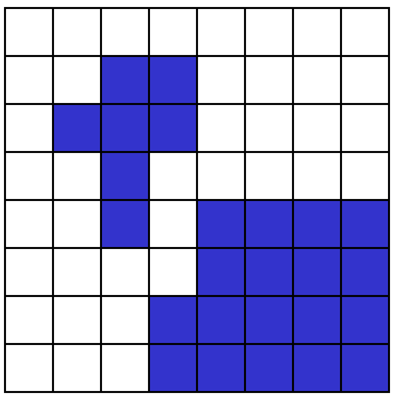
Quadtrees

- Grid with 2^k times 2^k pixels
- Depth is k +1
- Internal nodes always have 4 children
- Internal nodes represent a non-homogeneous region
- Leaves represent a homogeneous region and store the common value (or name)

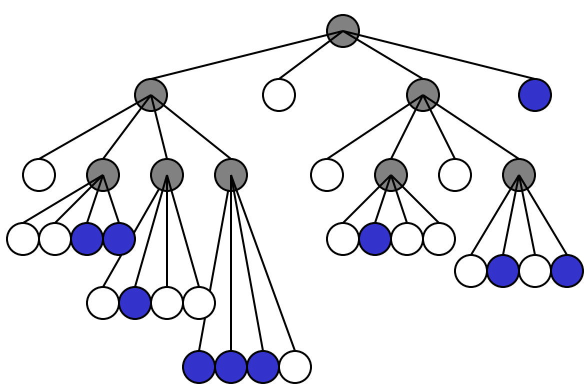
Quadtree complexity theorem

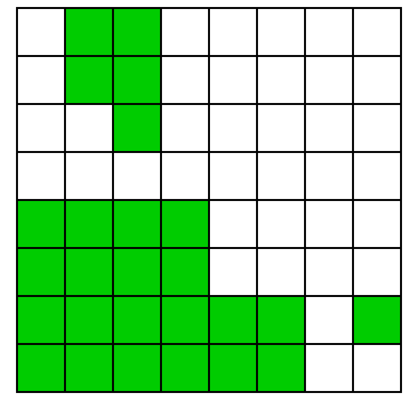
- A subdivision with boundary length *r* pixels in a grid of 2^k times 2^k gives a quadtree with $O(k \cdot r)$ nodes.
- Idea: two adjacent, different pixels “cost” at most 2 paths in the quadtree.

Overlay with quadtrees

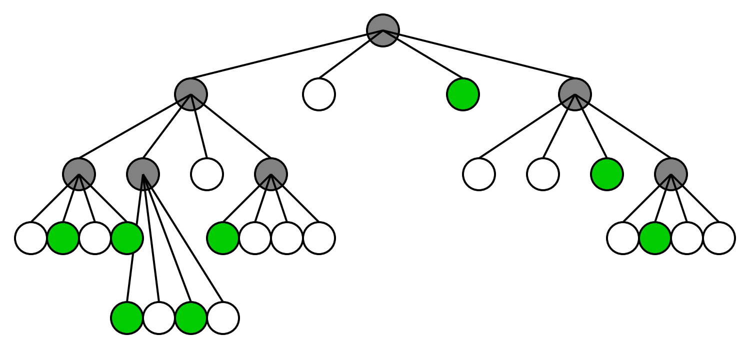


Water

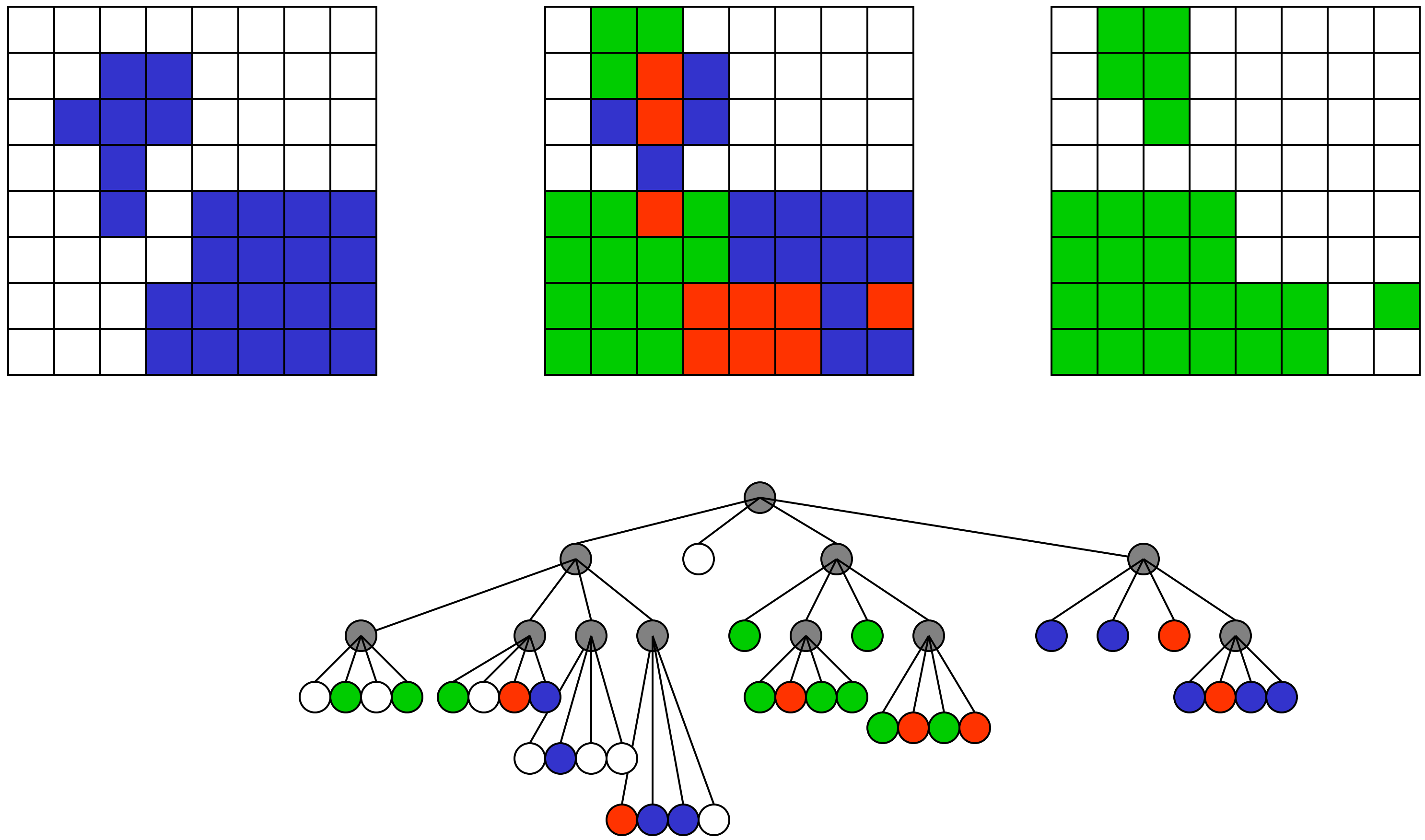




Acid rain with PH below 4.5

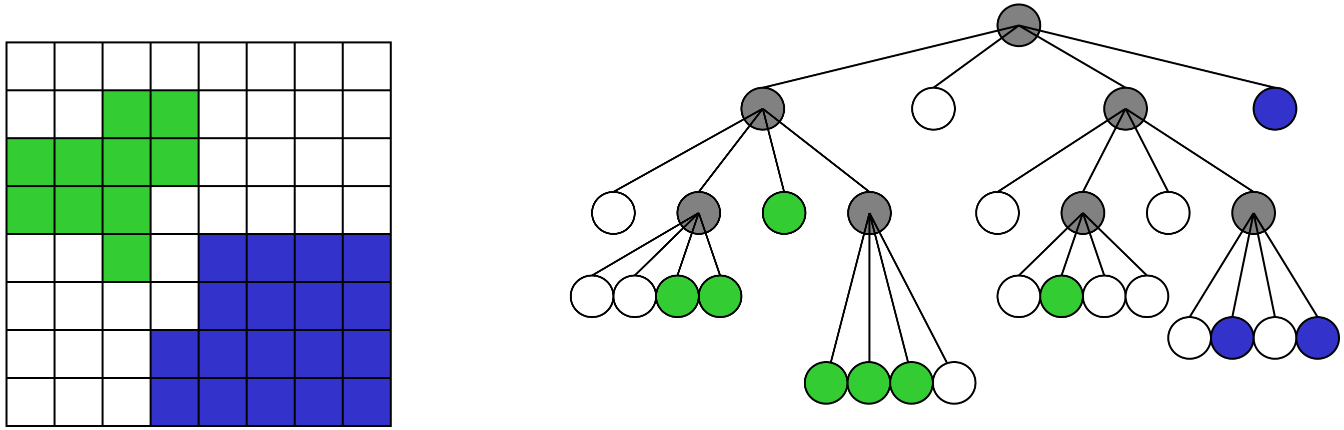


Result of overlay



Various queries

- Point location: trivial
- Windowing: descend into subtree(s) that intersect query window
- Traversal boundary polygon: up and down in the quadtree



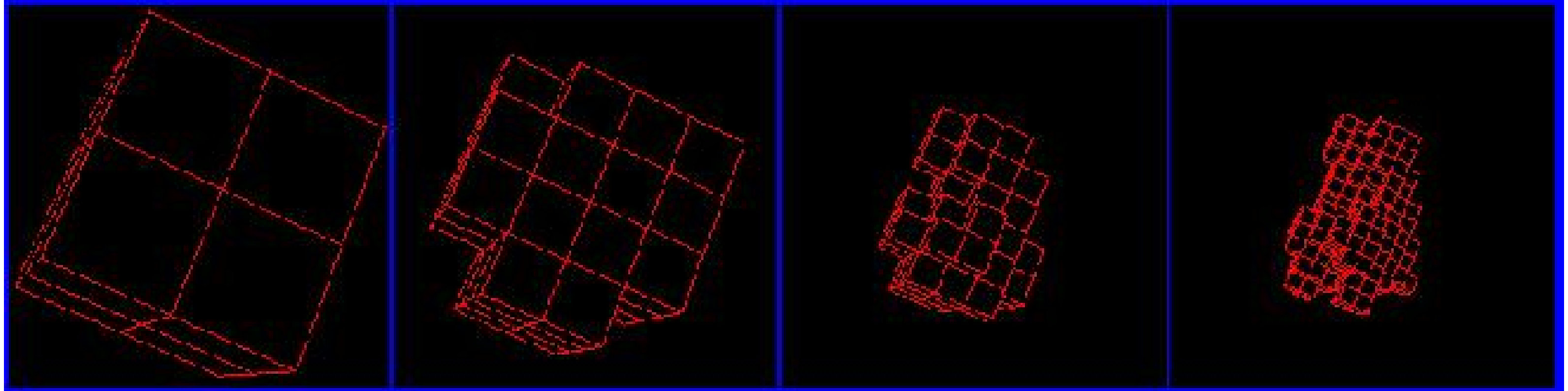
Octrees

- Like quadtrees, but for 3D applications.
- Breaks 3D space into octants
- Useful in graphics for representing 3D objects at different resolutions

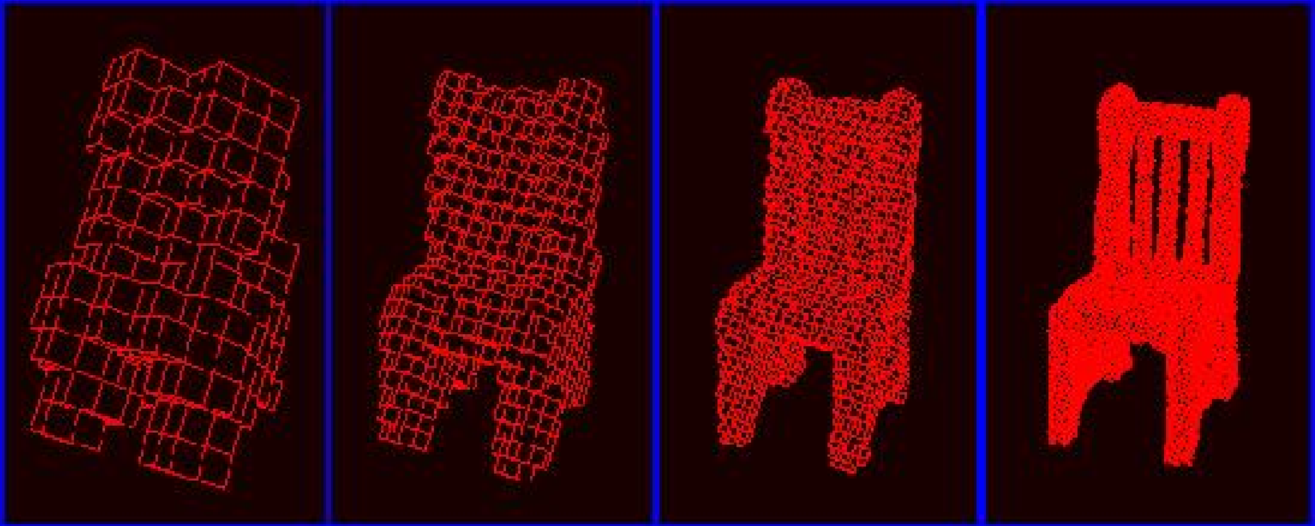
Hierarchical space carving

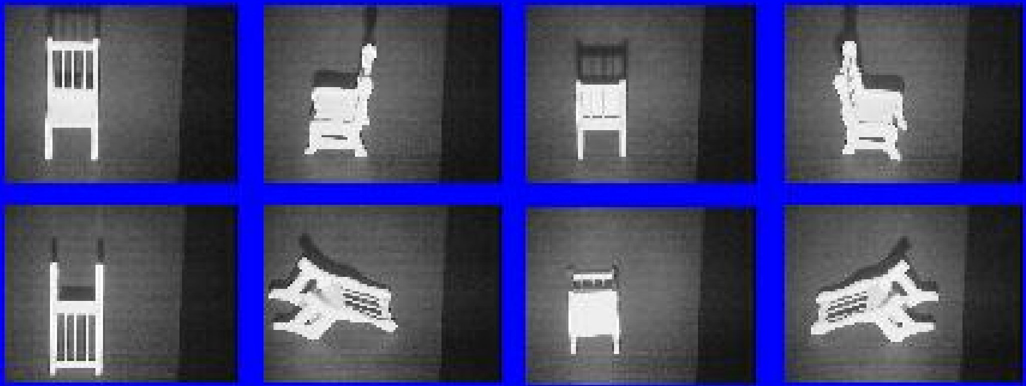
- Big cubes => fast, poor results
- Small cubes => slow, more accurate results
- Combination = octrees

RULES:
- cube's out => done
- cube's in => done
- else => recurse

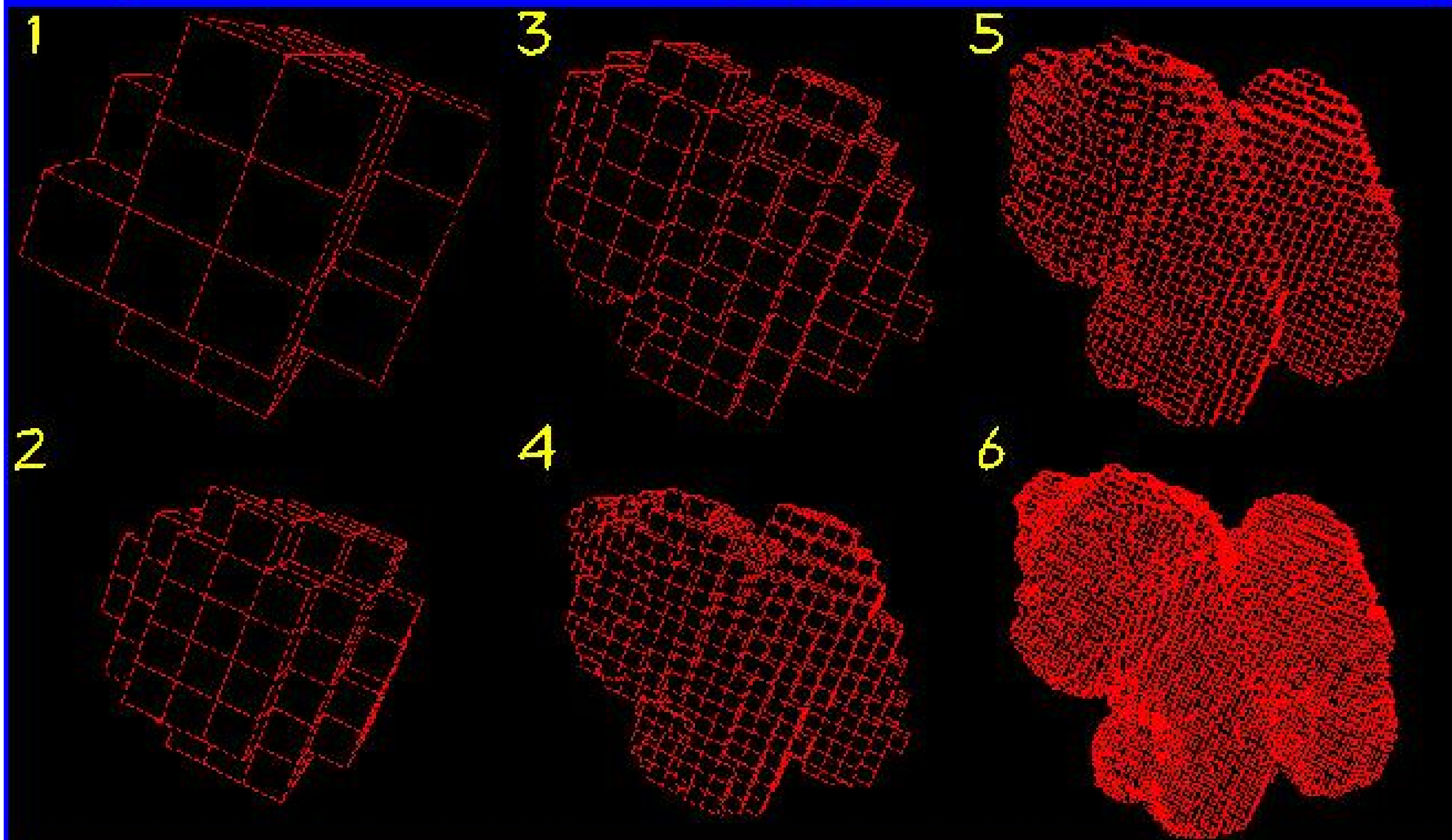


The rest of the chair

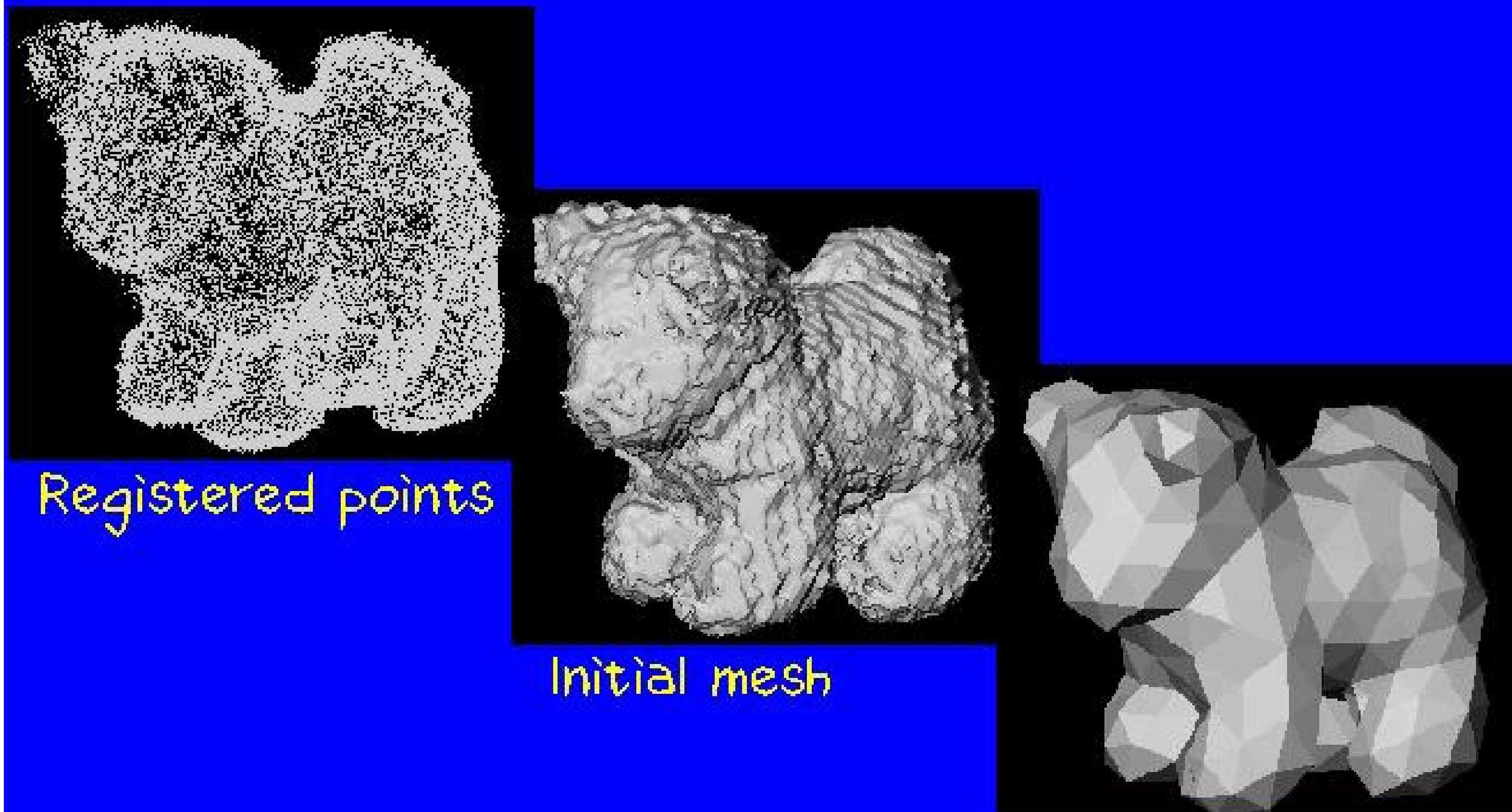




Same for a husky pup



Optimizing the dog mesh

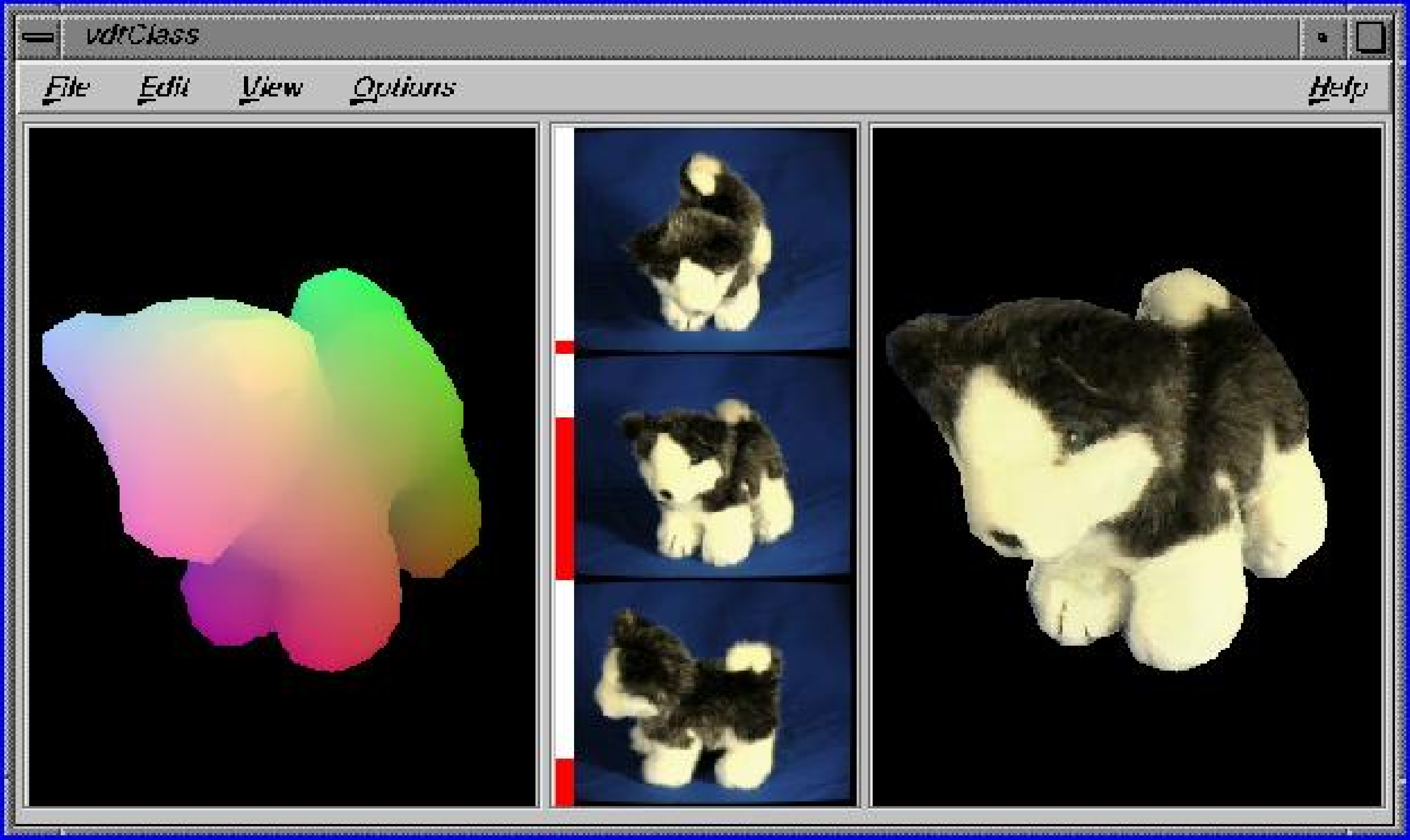


Registered points

Initial mesh

Optimized mesh

Our viewer



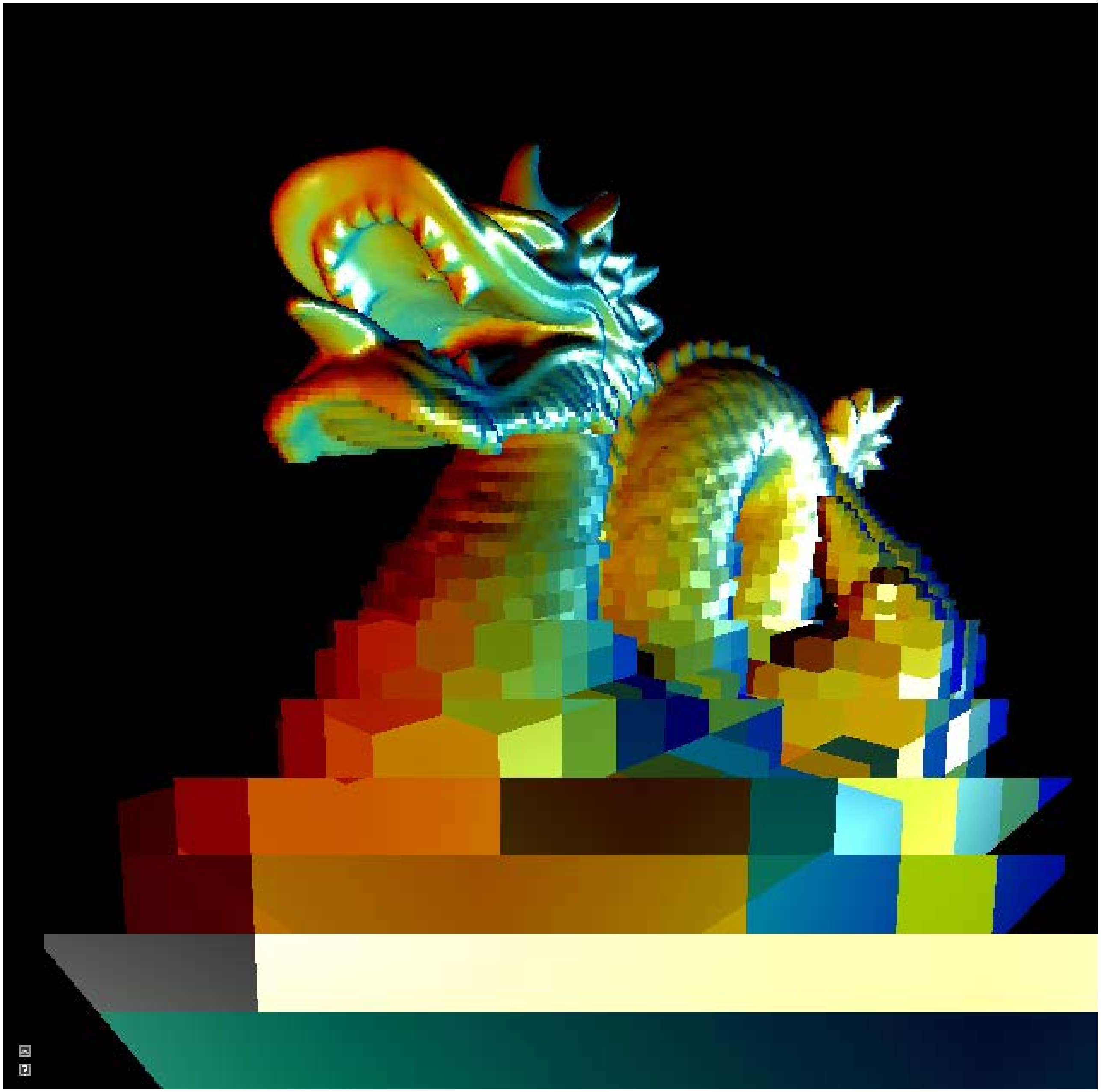
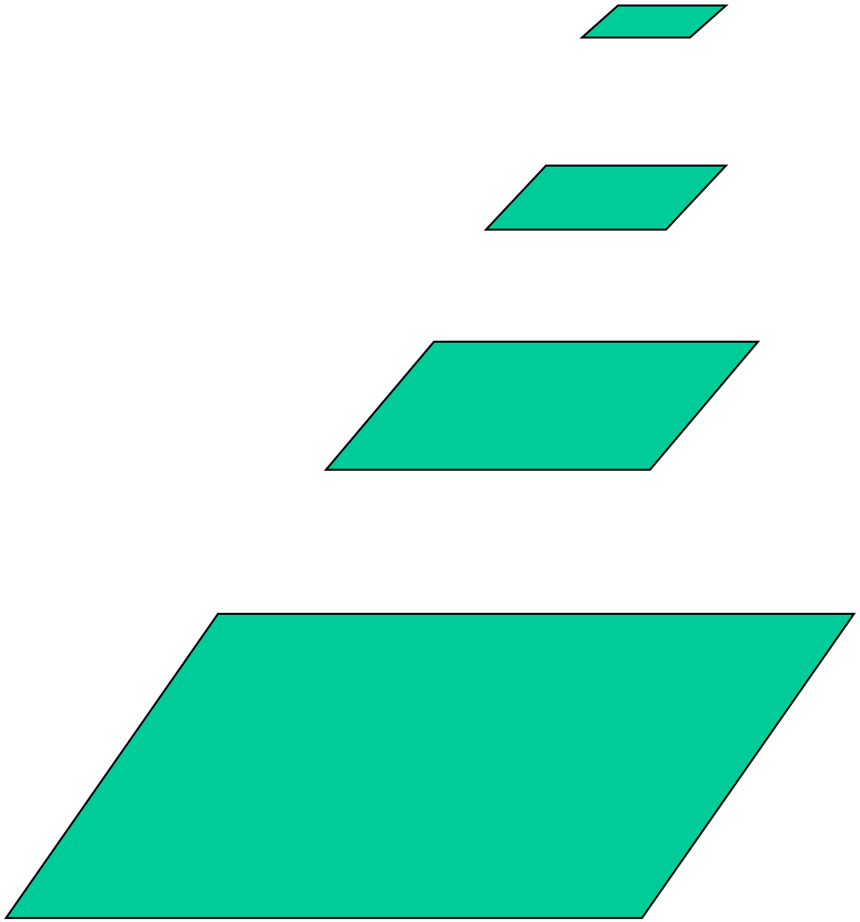


Image Pyramids



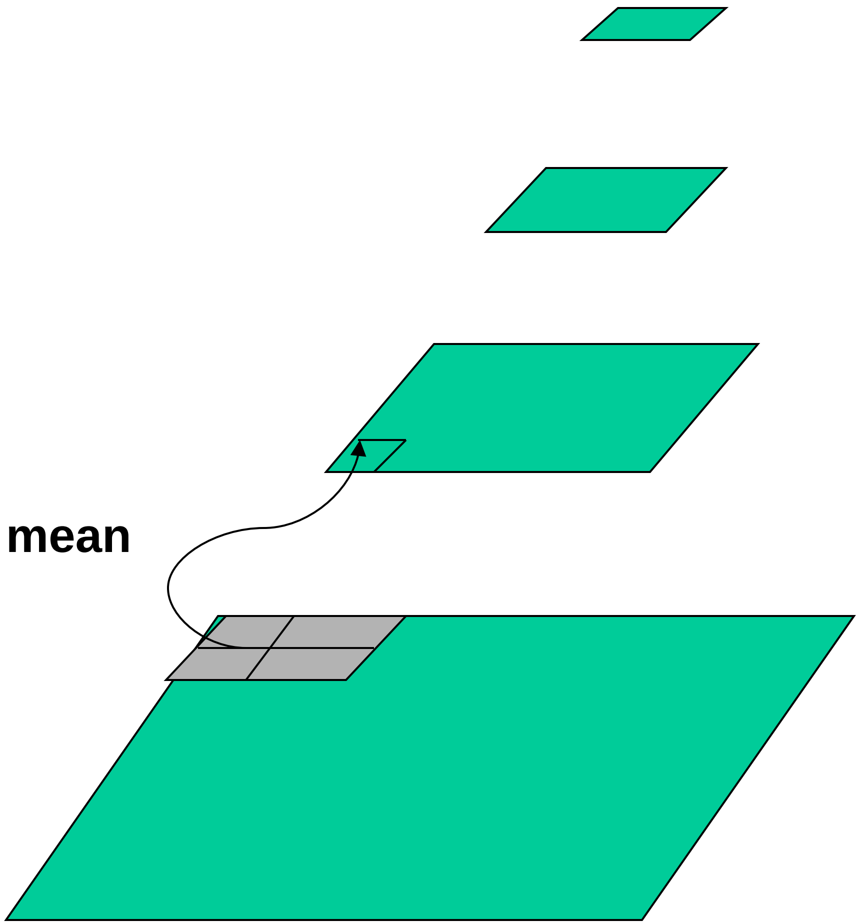
And so on.

3rd level is derived from the 2nd level according to the same funtion

2nd level is derived from the original image according to some function

Bottom level is the original image.

Mean Pyramid



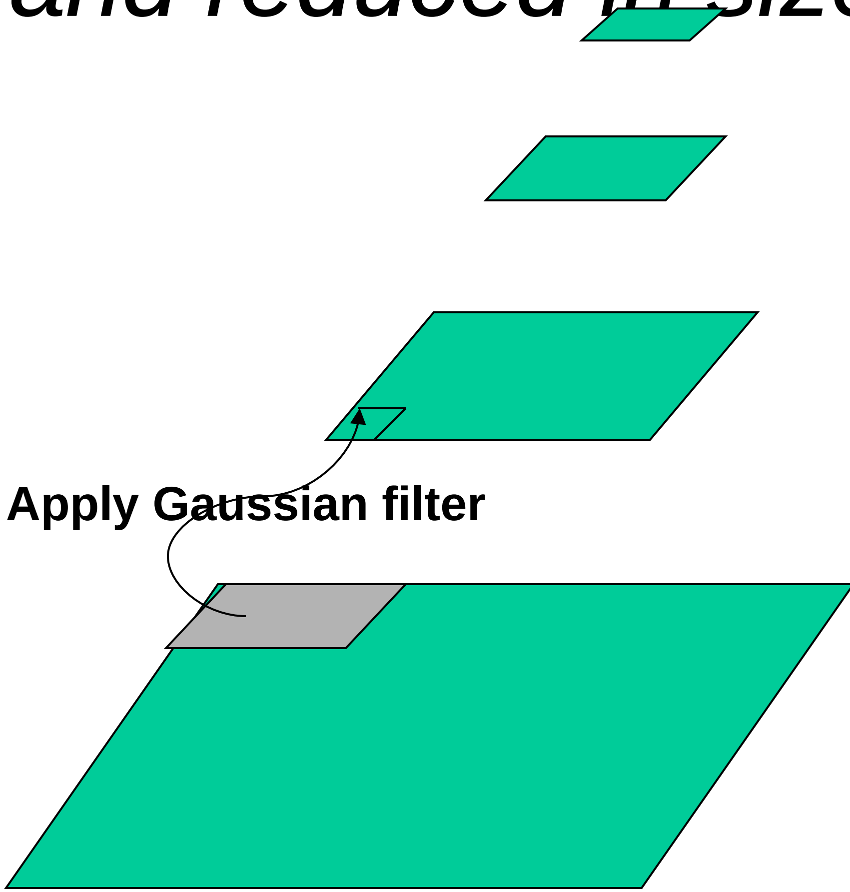
And so on.

At 3rd level, each pixel is the mean of 4 pixels in the 2nd level.

At 2nd level, each pixel is the mean of 4 pixels in the original image.

Bottom level is the original image.

Gaussian Pyramid
At each level, image is smoothed and reduced in size.

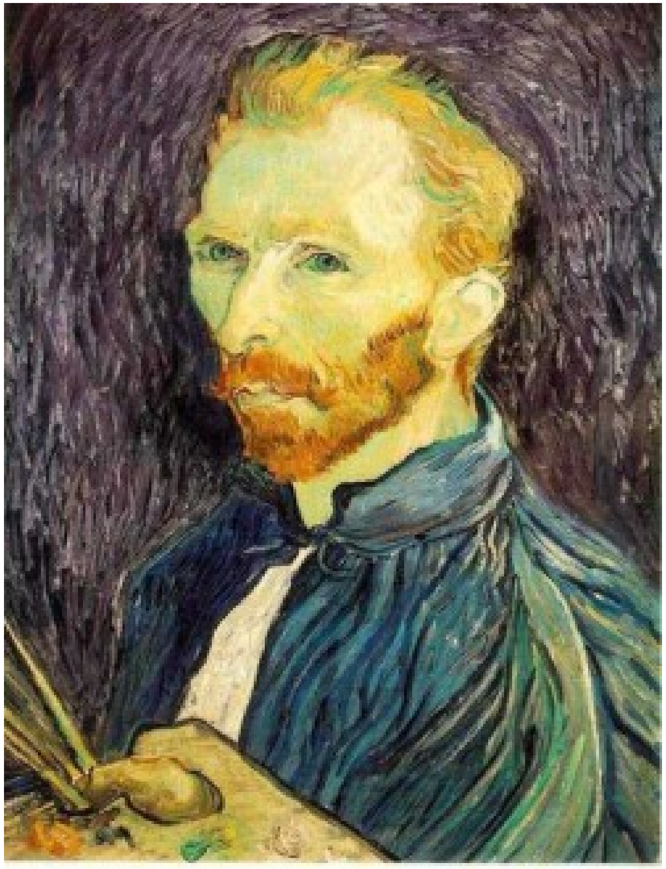


And so on.

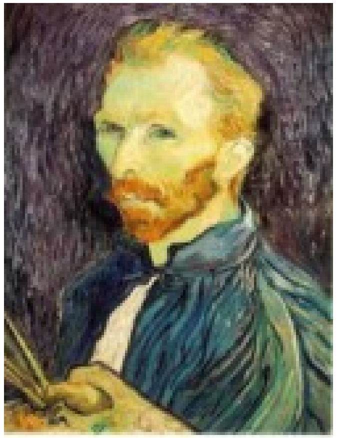
At 2nd level, each pixel is the result of applying a Gaussian mask to the first level and then subsampling to reduce the size.

Bottom level is the original image.

Example: Subsampling with Gaussian pre-filtering



Gaussian 1/2



G 1/4

G 1/8

Backtracking Search in AI/Vision

- Start at the root of a search tree at a “state”
- Generate children of that state
- For each child
  - If the child is the goal, done
  - If the child does not satisfy the constraints of the problem, ignore it and keep going in this loop
  - Else call the search recursively for this child
- Return

This is called **backtracking,** because if it goes through all children of a node and finds no solution, it returns to the parent and continues with the children of that parent.

Graph Matching

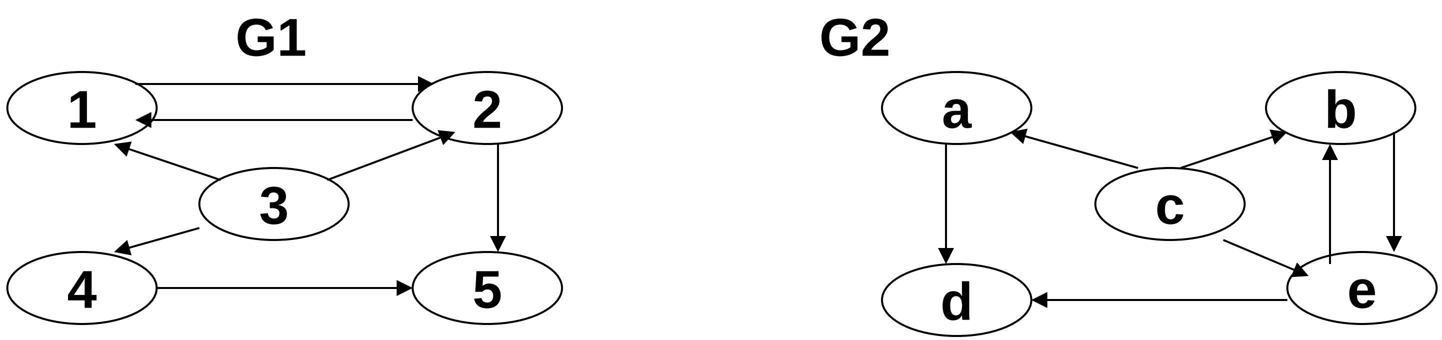
Input: 2 digraphs G1 = (V1,E1), G2 = (V2,E2)

Questions to ask:

1. **Are G1 and G2 isomorphic?**
2. **Is G1 isomorphic to a subgraph of G2?**
3. **How similar is G1 to G2?**
4. **How similar is G1 to the most similar subgraph of G2?**

Isomorphism for Digraphs

G1 is isomorphic to G2 if there is a 1-1, onto mapping $h: V1 \rightarrow V2$ such that $(v_i, v_j) \in E1$ iff $(h(v_i), h(v_j)) \in E2$.



Find an isomorphism $h: \{1,2,3,4,5\} \rightarrow \{a,b,c,d,e\}$.
Check that the condition holds for every edge.

Answer: h(1)=b, h(2)=e, h(3)=c, h(4)=a, h(5)=d

Isomorphism for Digraphs

G1 is isomorphic to G2 if there is a 1-1, onto mapping h: V1 → V2 such that (vi,vj) ∈ E1 iff (h(vi), h(vj)) ∈ E2

Answer: h(1)=b, h(2)=e, h(3)=c, h(4)=a, h(5)=d
(1,2) ∈ E1 and (h(1),h(2))=(b,e) ∈ E2.
(2,1) ∈ E1 and (e,b) ∈ E2.
(2,5) ∈ E1 and (e,d) ∈ E2.
(3,1) ∈ E1 and (c,b) ∈ E2.
(3,2) ∈ E1 and (c,e) ∈ E2.
...

Subgraph Isomorphism for Digraphs

G1 is isomorphic to a subgraph of G2 if there is a 1-1 mapping h: V1 $\rightarrow$ V2 such that (vi,vj) $\in$ E1 $\Rightarrow$ (h(vi), h(vj)) $\in$ E2.

Isomorphism and subgraph isomorphism are defined similarly for undirected graphs.

In this case, when (vi,vj) $\in$ E1, either (vi,vj) or (vj,vi) can be listed in E2, since they are equivalent and both mean {vi,vj}.

Subgraph Isomorphism for Graphs

G1 is isomorphic to a subgraph of G2 if there is a 1-1 mapping $h: V1 \rightarrow V2$ such that $\{vi,vj\} \in E1 \Rightarrow \{ h(vi), h(vj) \} \in E2$.

Because there are no directed edges, there are more possible mappings.

1	2	3	
c	b	d	
c	d	b	(shown on graph)
b	c	d	
b	d	c	
d	b	c	
d	c	b	

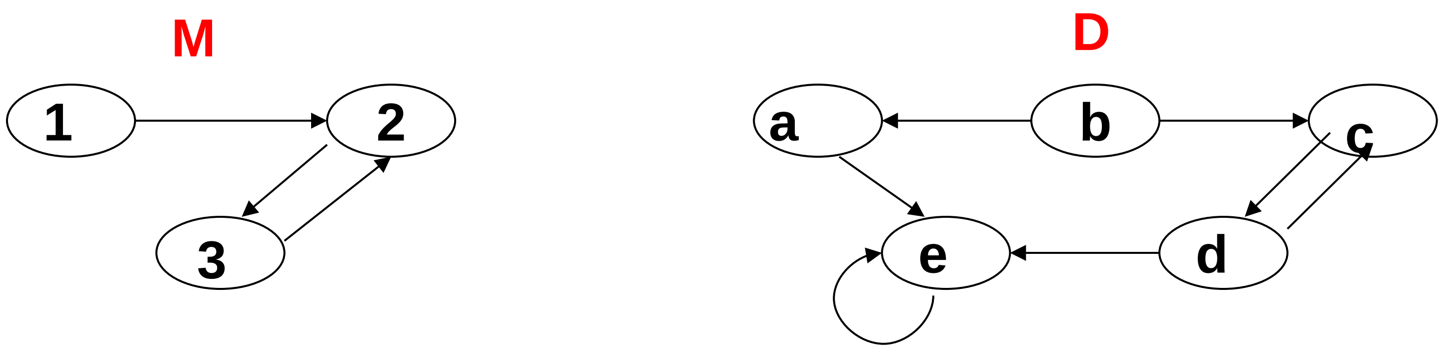
Graph Matching Algorithms: Subgraph Isomorphism for Digraph

Given model graph M = (VM,EM)
data graph D = (VD,ED)

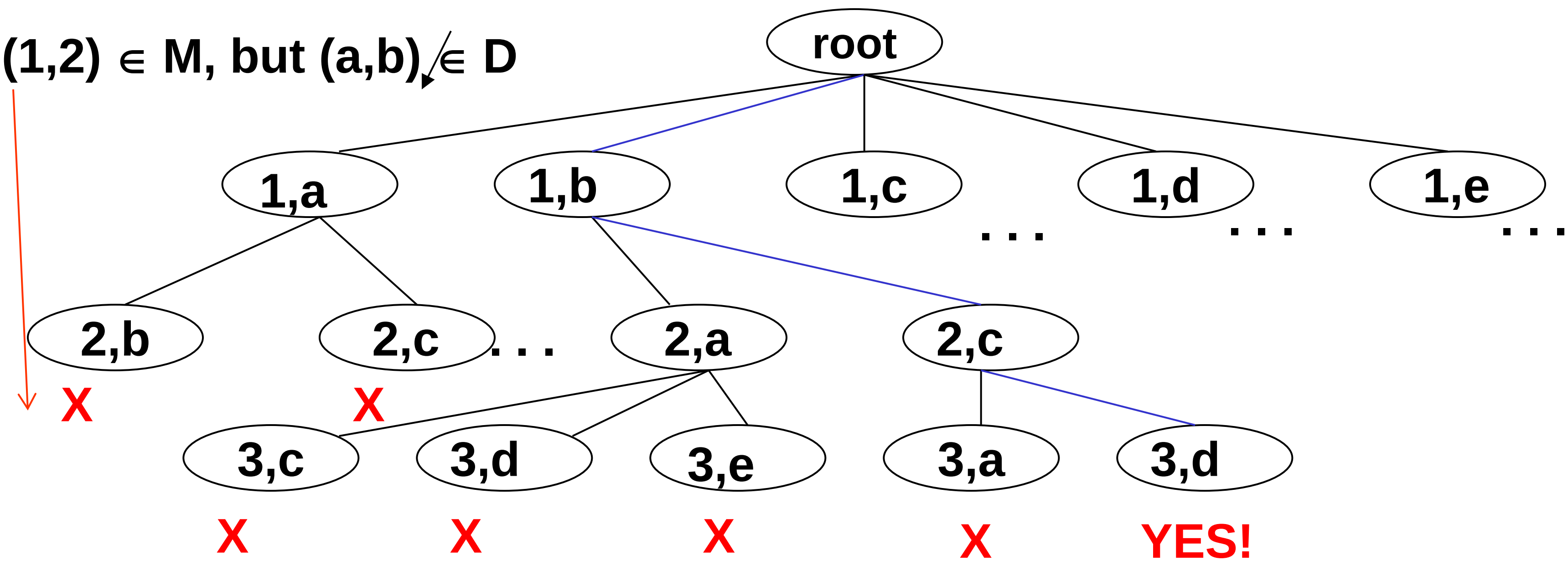
Find 1-1 mapping h:VM $\rightarrow$ VD

satisfying (vi,vj) $\in$ EM $\Rightarrow$ ((h(vi),h(vj)) $\in$ ED.

Method: Recursive Backtracking Tree Search (Order is depth first, leftmost child first.)

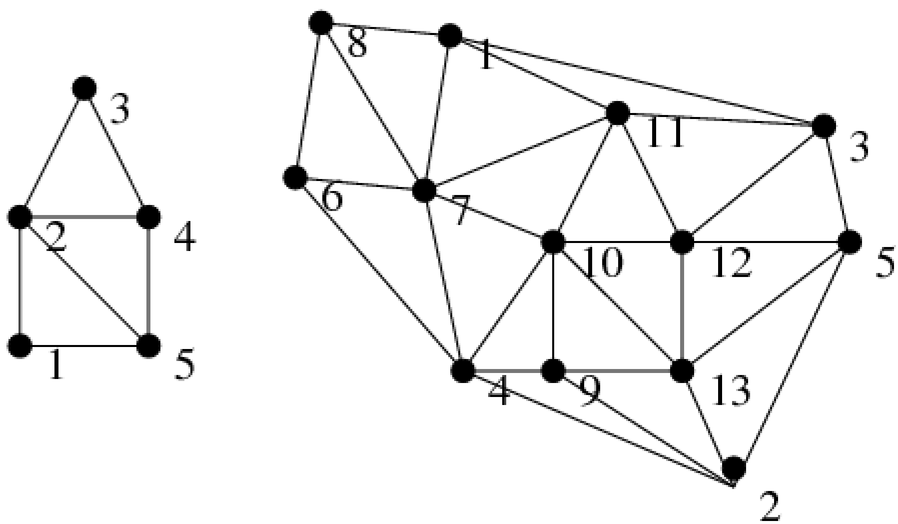


(1,2) $\in$ M, but (a,b) $\notin$ D

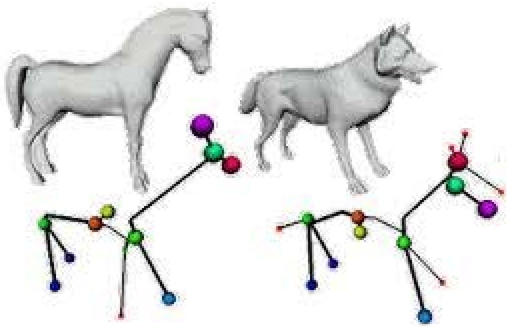


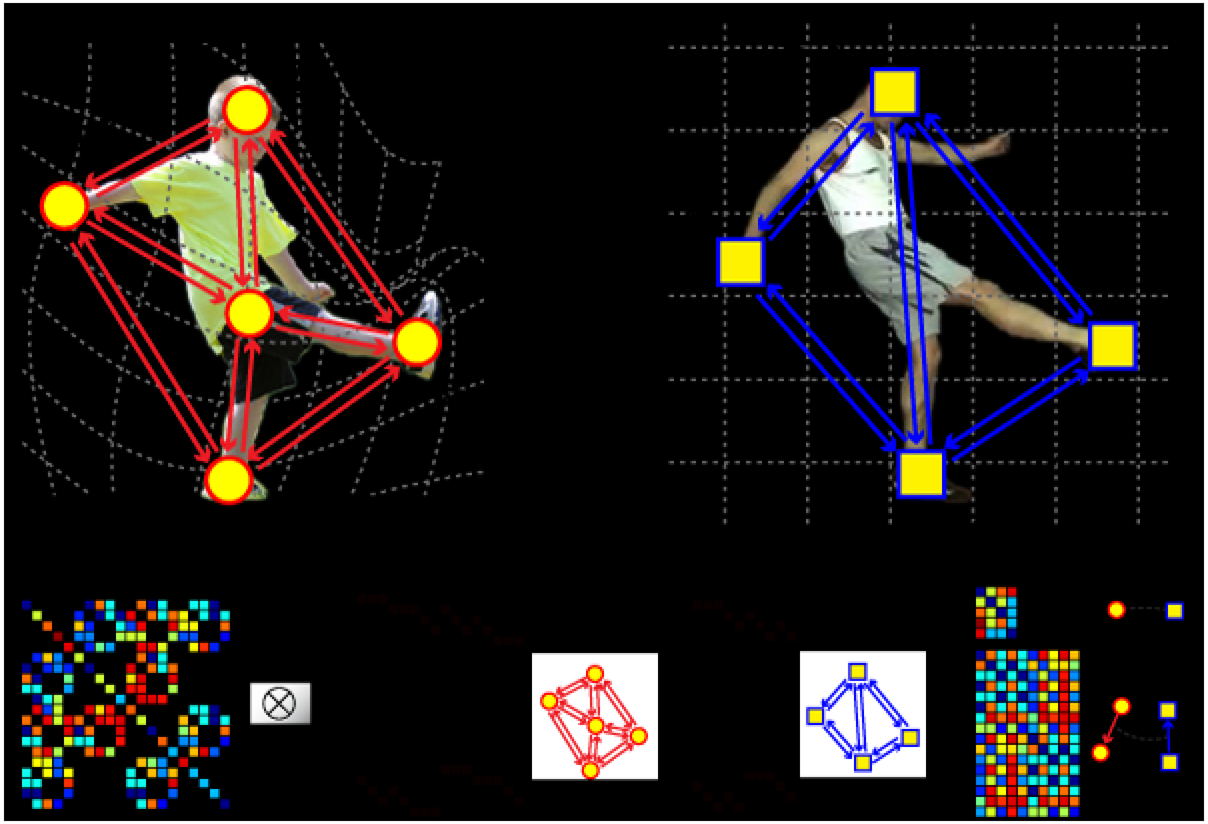
Application to Computer Vision

Find the house model in the image graph.



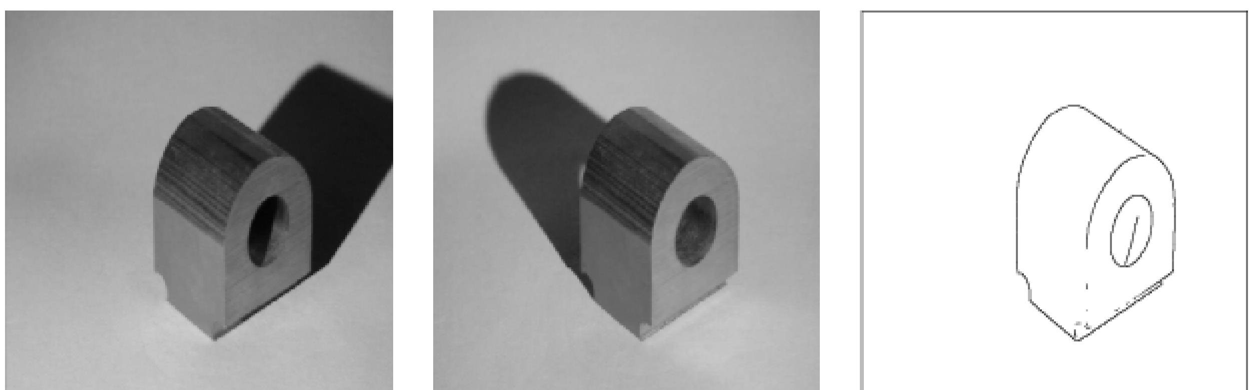
More Examples





RIO: Relational Indexing for Object Recognition

- **RIO worked with industrial parts that could have**
  - planar surfaces
  - cylindrical surfaces
  - threads



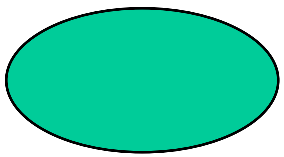
Object Representation in RIO

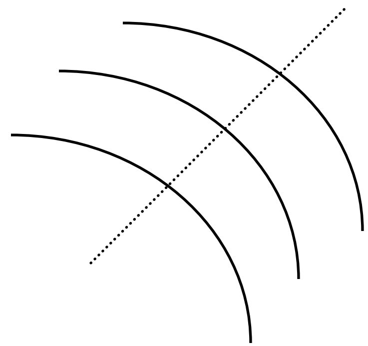
- **3D objects are represented by a 3D mesh and set of 2D view classes.**

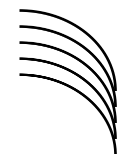
- **Each view class is represented by an attributed graph whose nodes are features and whose attributed edges are relationships.**

- **Graph matching is done through an indexing method, not covered here.**

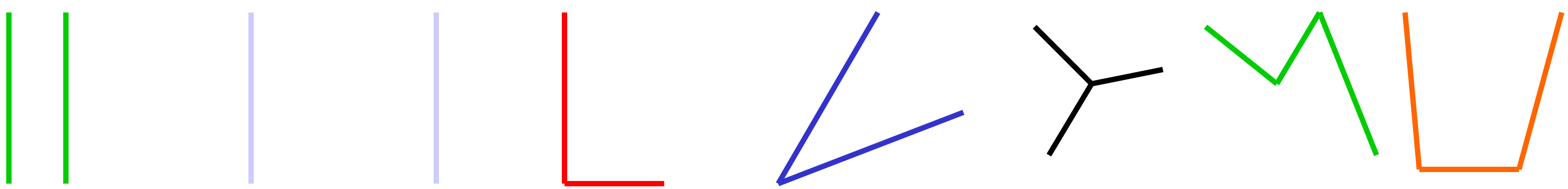
RIO Features







ellipses **coaxials** **coaxials-multi**



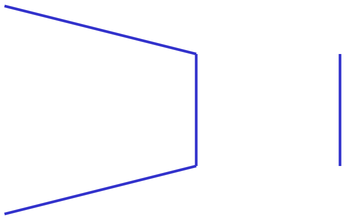
parallel lines close and far

junctions

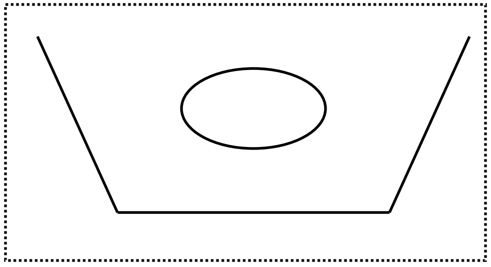
triples

L **V** **Y** **Z** **U**

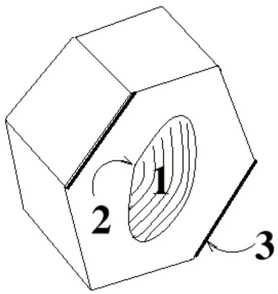
RIO Relationships



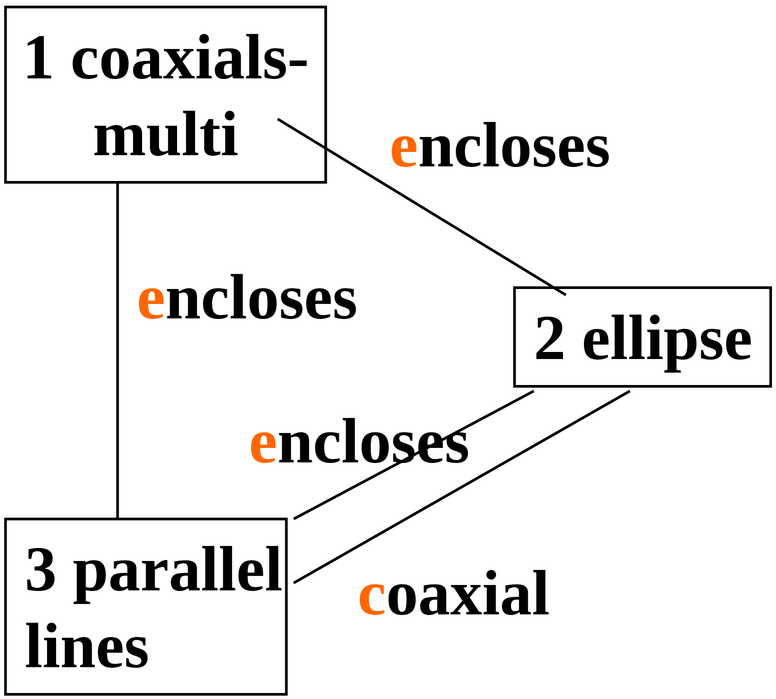
- **share one arc**
- **share one line**
- **share two lines**
- **coaxial**
- **close at extremal points**
- **bounding box encloses / enclosed by**



MODEL-VIEW

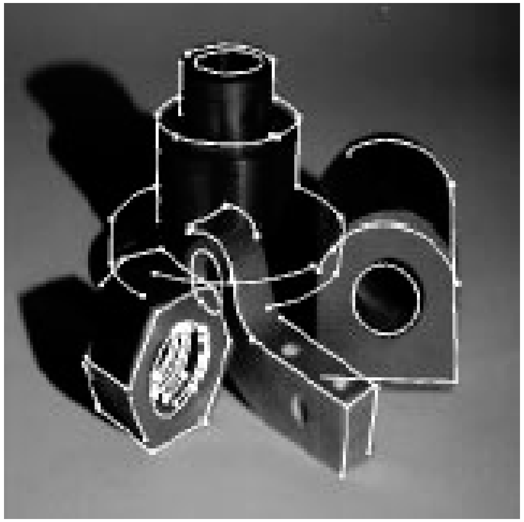


Graph Representation

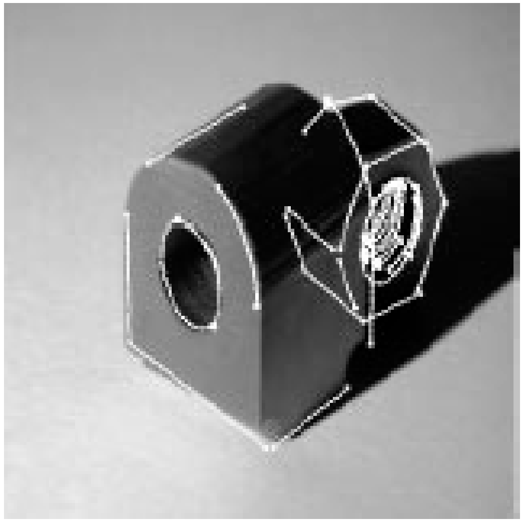


This is just a piece of the whole graph.

Sample Alignments
3D to 2D Perspective Projection



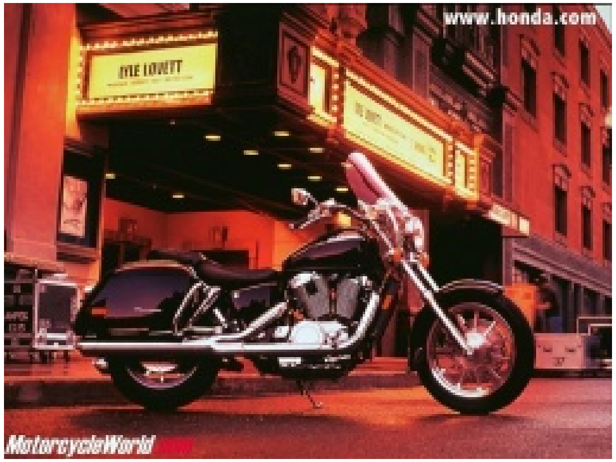
(a)



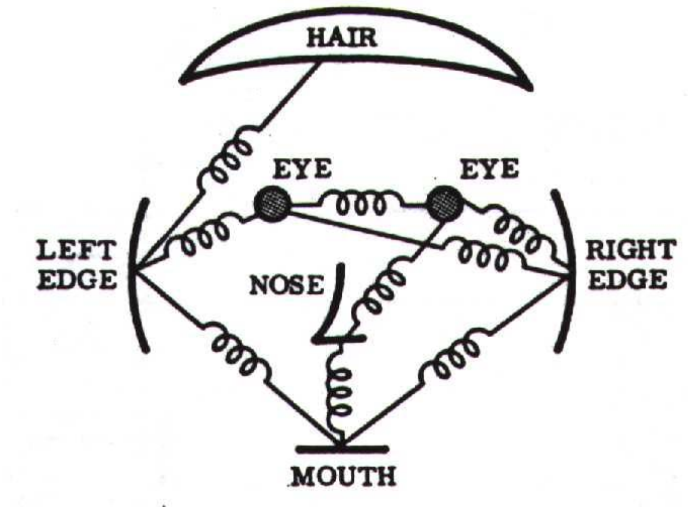
(b)

Fergus Object Recognition by Parts:

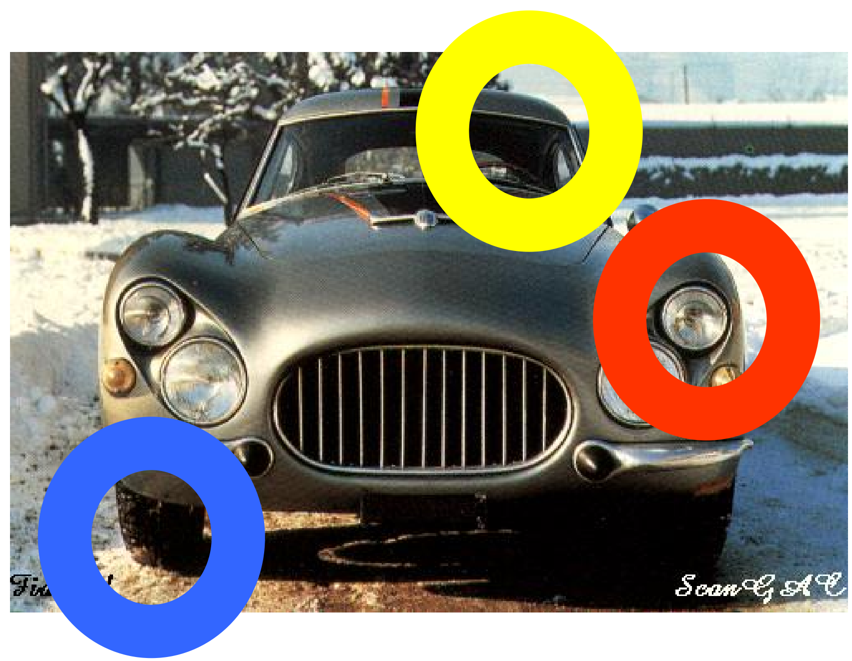
- Enable Computers to Recognize Different Categories of Objects in Images.



Model: Constellation Of Parts

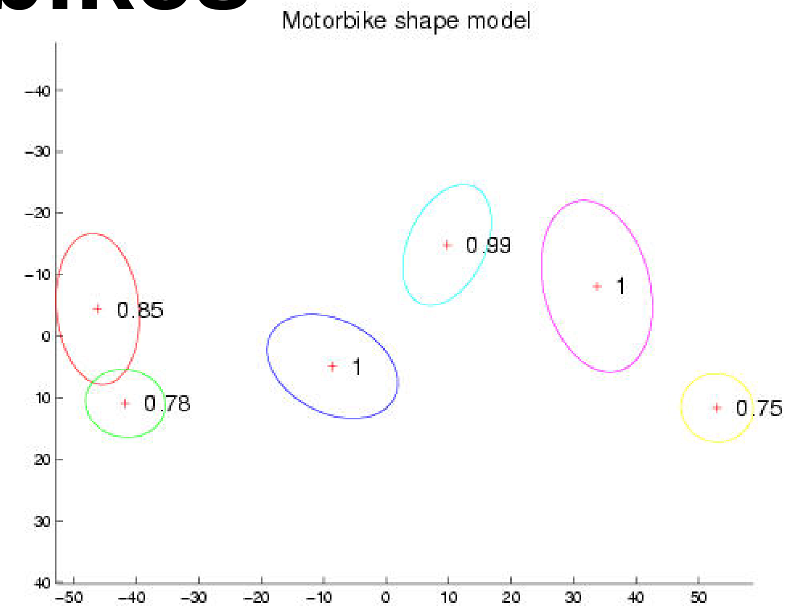


Fischler & Elschlager, 1973



Motorbikes

Part 1 – Det:5e–18
Part 2 – Det:8e–22
Part 3 – Det:6e–18
Part 4 – Det:1e–19
Part 5 – Det:3e–17
Part 6 – Det:4e–24
Background – Det:5e–19



Motorbike shape model
0.85
0.78
1
0.99
1
0.75
–40
–30
–20
–10
0
10
20
30
40
–50
–40
–30
–20
–10
0
10
20
30
40
50