Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät
Fachbereich Informatik
Arbeitsbereich SAV/BV (KOGS)

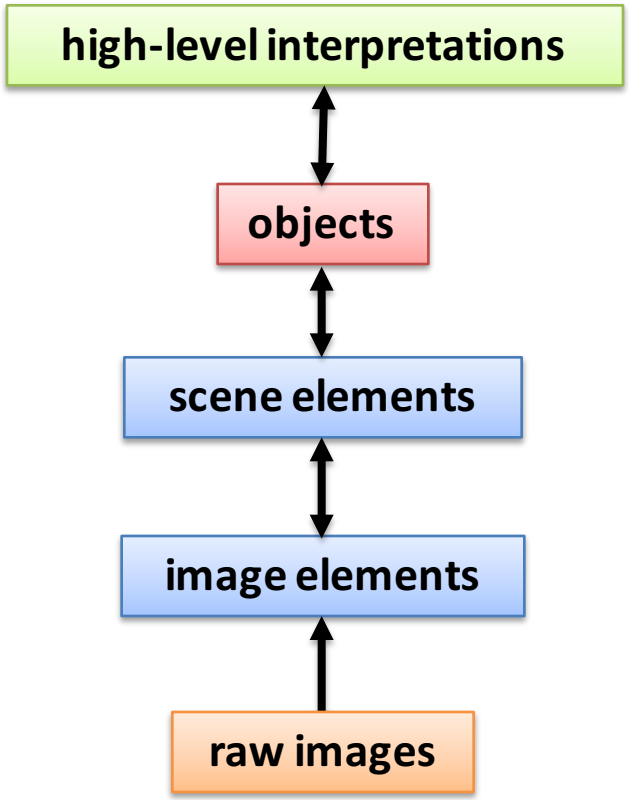
Image Processing 1 (IP1)
Bildverarbeitung 1

Lecture 21: Object Recognition 1

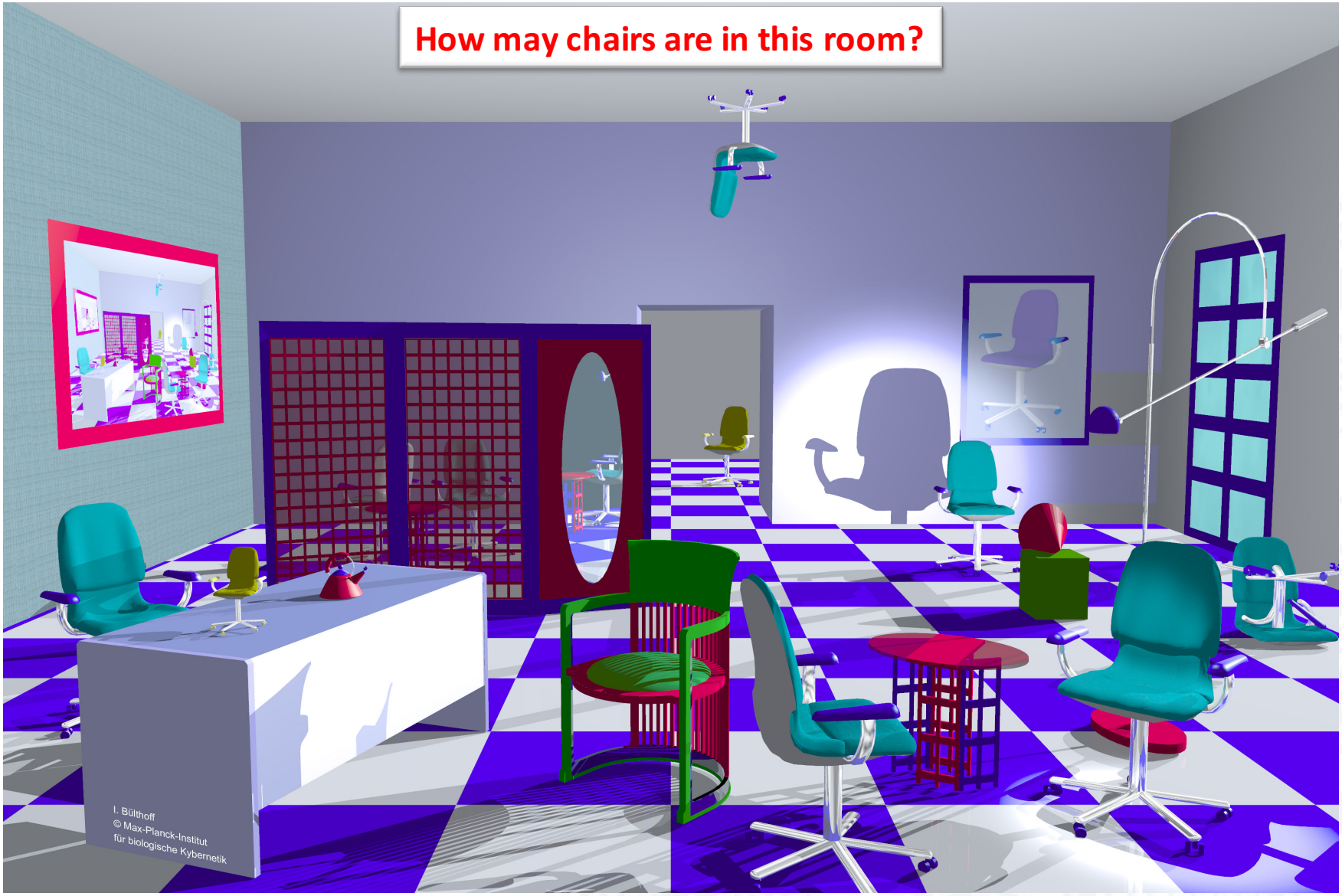
Winter Semester 2015/16

Slides: Prof. Bernd Neumann
Slightly revised by: Dr. Benjamin Seppke & Prof. Siegfried Stiehl

Object Recognition



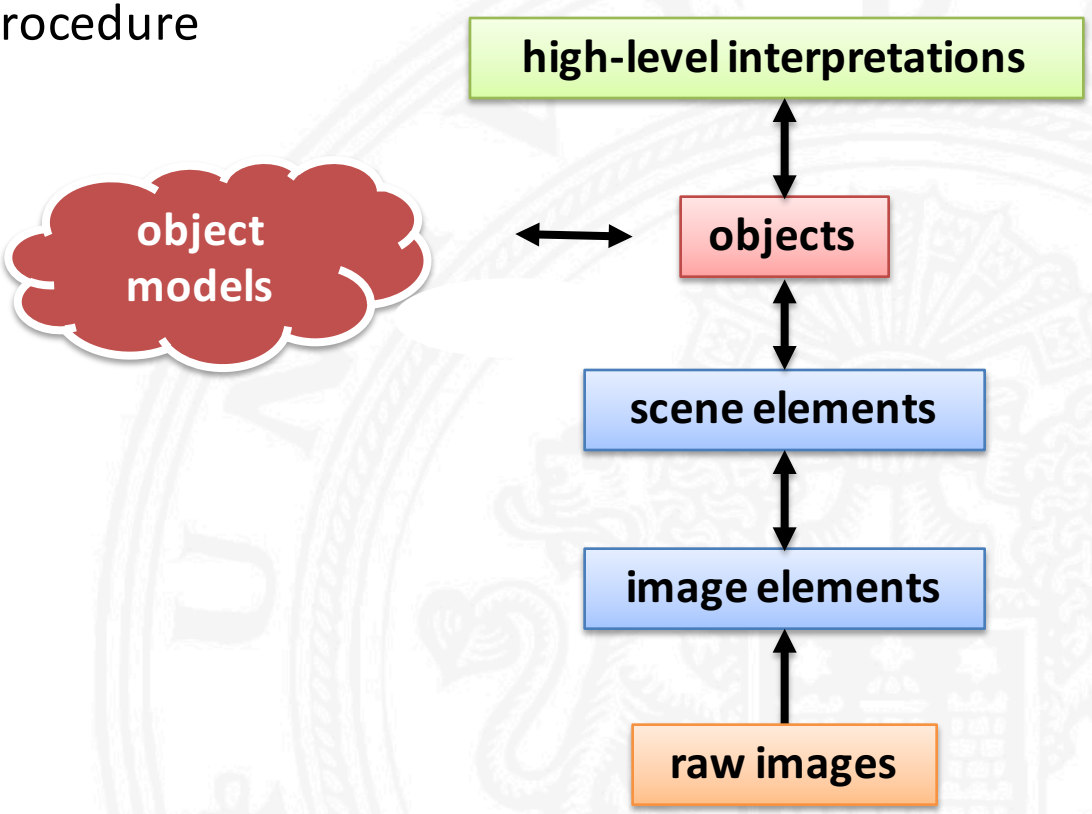
- object recognition is a typical goal of image analysis
- object recognition includes:
  - object identification
    recognizing that one object instance is (physically) identical to another object instance
  - object classification
    assigning an object to one of a set of predetermined classes
  - object categorization
    assigning an object to an object category (as proposed in biological vision)


How may chairs are in this room?
I. Bülthoff
© Max-Planck-Institut
für biologische Kybernetik

Model-based Object Recognition

Recognizing objects using an

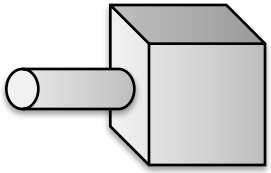
- explicit description ("model") and a
- generic recognition procedure



3D Models vs. 2D Models

1. Requirement:
   - Object models must represent invariant class properties

   → 3D models, properties independent of views

e.g. 

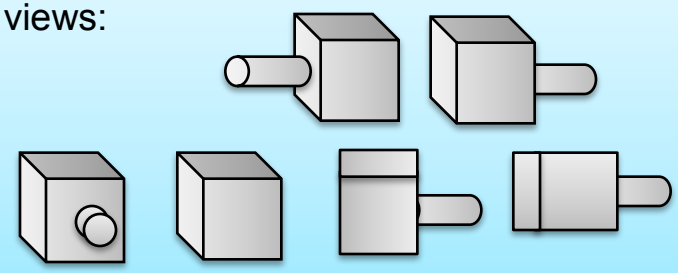
class hammer
- is-a aggregat
- has-parts part1, part2
- is-a part1 cube
- is-a part2 cylinder
- coaxially-connected part1 part2

2. Requirement:
   - Object models must support recognition

   → 2D models, view-dependent properties

Modern approaches to object recognition are typically a compromise of Requirements 1 and 2.

class hammer
views:

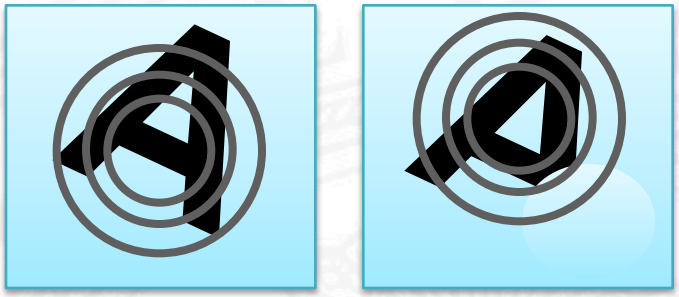


Holistic Models vs. Component Models

Holistic ("global") models:

- properties refer to complete object
- local disturbances may jeopardize all properties

e.g. area, polar signature, NN classifier



Component models:

- object model is described by components and relations between components
- properties refer to individual components
- local disturbances affect only local properties

Example of components:

3D Shape Models

Several 3D shape models have been developed for engineering applications:

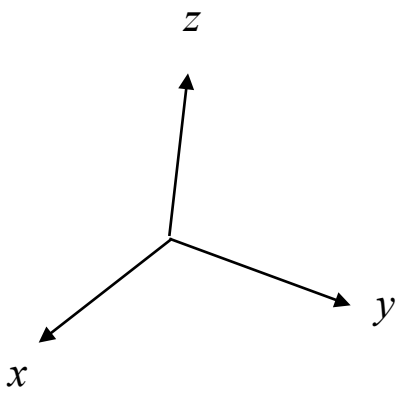
- 3D space occupancy
- Oct-trees
- CSG ("Constructive Solid Geometry") models
- 3D surface triangulation

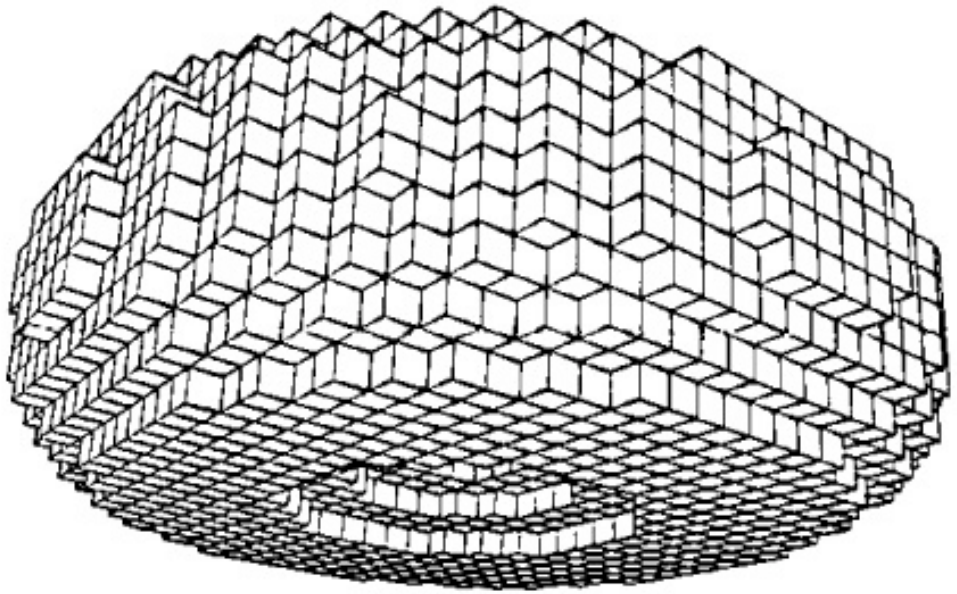
In general, pure 3D models are not immediately useful for Computer Vision because they do not support recognition.

In support of recognition, special 3D models have been developed which include view-related information:

- EGI ("Extended Gaussian Image")
- Generalized cylinders

3D Space Occupancy Model



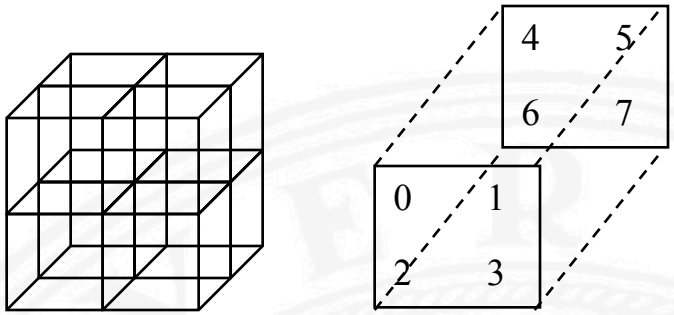


3D shape represented by cube primitives

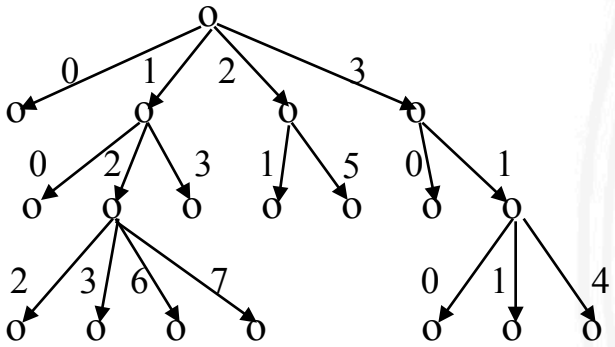
- useful for highly irregular shapes (e.g. medical domain)
- useful for robotics applications (e.g. collision avoidance)
- interior cubes do not provide information relevant for views
- no explicit surface properties (e.g. surface normals)

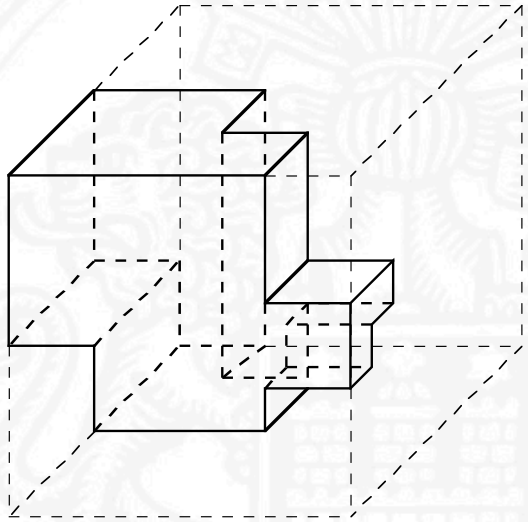
Oct-trees

- hierarchical 3D shape model
- analog to 2D quad-trees
- each cube is recursively decomposed into 8 subcubes
- access via numbering code



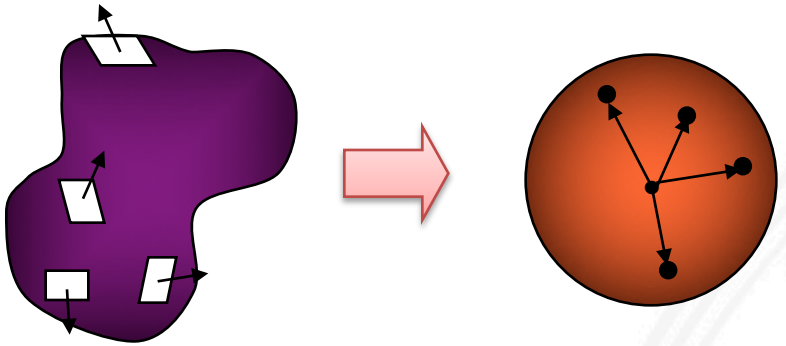
Oct-tree for example (right):





Extended Gaussian Image (EGI)

- 3D shape model based on a surface slope histogram
- extended to provide view-point information for recognition



example of a 3D surface

entries on Gaussian sphere

B.K.P. Horn
Robot Vision
The MIT Press 1986

Each entry represents information for a particular 3D slope and viewing direction:

1. quotient of surface area with this slope and total surface area
2. quotient of visible 3D surface area and area of its 2D projection (as viewed from this direction)
3. direction of axis of minimal inertia of 2D projection of visible surface (as viewed from this direction)

Recognition with EGI Models

Properties of EGIs:

- scale invariant
- rotation of object corresponds to equivalent rotation of EGI
- convex shapes can be uniquely reconstructed
- In particular: A convex polyhedron can be reconstructed from the set of orientations and associated areas $\{(o_1\ a_1)\ (o_2\ a_2)\ \ldots\ (o_N\ a_N)\}$
- In general, reconstruction requires an iterative algorithm

Recognition procedure:

It is assumed that 3D surface normals are determined (e.g. by laser measurements)

- determine direction of axis of minimal inertia
- determine projected surface area
- determine patches of (approximately) constant 3D surface inclination
- constrained search for models which match the measurements

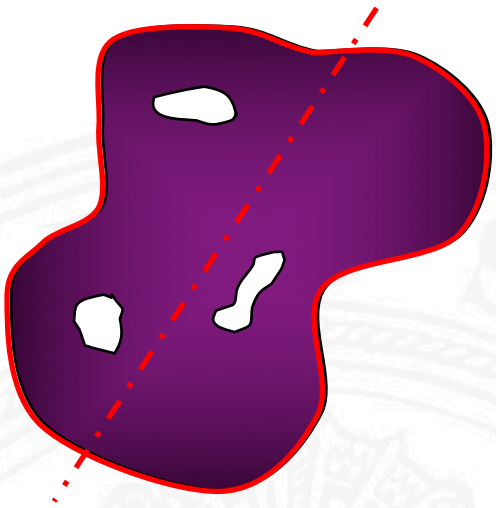
Illustration of EGI Recognition Procedure

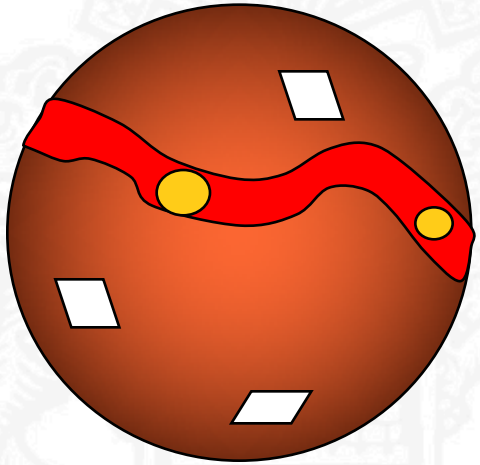
1. Determine direction of axis of minimal inertia
   → locations on EGI with corresponding entries

2. Determine projected surface area
   → subset of locations determined by 1)

3. Determine patches of constant 3D surface inclination
   → rotate EGI into viewing direction of 1) and 2), compare surface area with corresponding entries

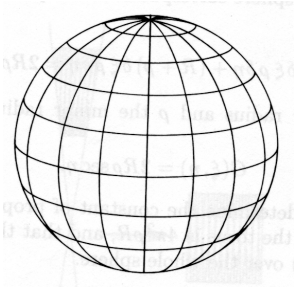
4. Constrained search for models which match the measurements
   → if 1) to 3) do not match, choose other models



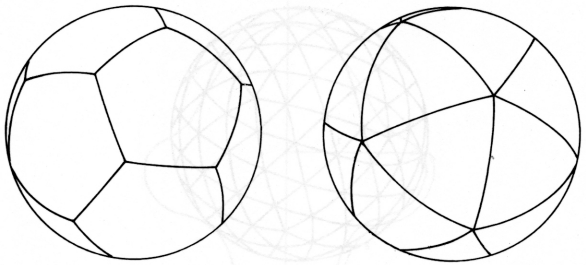


Discretizing the Surface of a Sphere

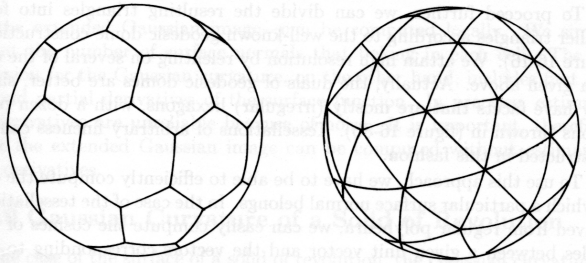
For a computer representation of an Extended Gaussian Sphere, the surface of the sphere has to be tessellated into patches of approximately equal size.



geodesic tessellation has undesirable properties



dodecahedron (left) and icosahedron (right) provide tessallations with 12 and 20 cells, respectively



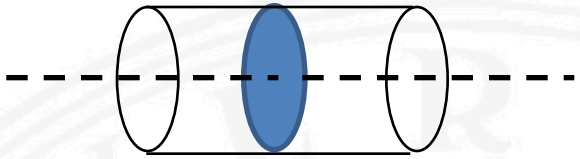
truncated icosahedron (left) provides 12 pentagonal and 20 hexagonal faces, pentakis dodekahedron (right) provides 60 triangular faces

Further refinements can be obtained by triangularization within the cells of a regular or semiregular solid.

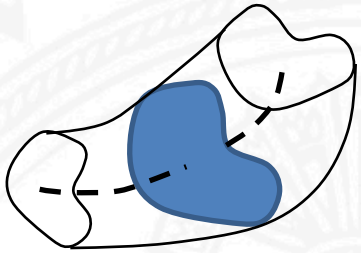
Generalized Cylinders

3D surface determined by sweeping a closed curve along a line

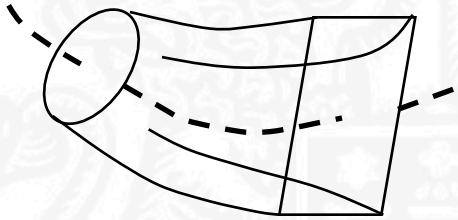
ordinary cylinder swept out by a circle along a straight line



generalized cone swept out by an arbitrary planar cross section, varying in size, along a smooth axis (Binford 71)

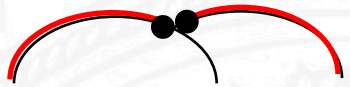


generalized cylinder swept out by a closed curve at an angle to a curved axis subject to a deformation function



Conditions for 3D Reconstruction from Contours

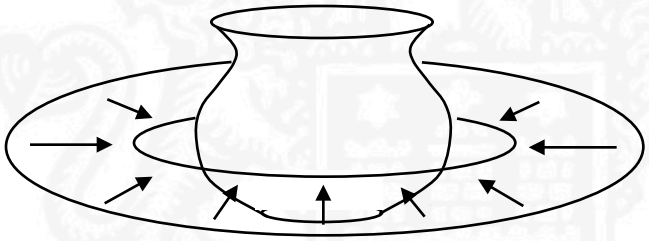
1. Each line of sight touches the body at a single point
   → we see a "contour generator"
2. Nearby points on the contour in the image are also nearby in 3D (with only few exceptions)



2 distant points projected onto nearby contour points

3. The contour generator is planar
   → hence inflections of the contour in 2D correspond to inflections in 3D

If a surface is smooth and if conditions 1 to 3 hold for all viewing directions in any plane, then the viewed surface is a generalized cone. (Marr 77)



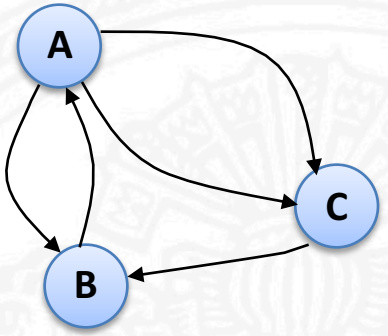
Relational Models

Relational models describe objects (object classes) based on parts (components) and relations between the parts

Relational model can be represented as structure with nodes and edges:

Nodes: parts with properties

e.g. 



Edges: relations between parts

e.g.

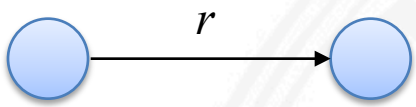
- **obtuse-angle**
- **2cm-distance**
- **touches**
- **surrounds**
- **left-of**
- **after**

Relations between Components

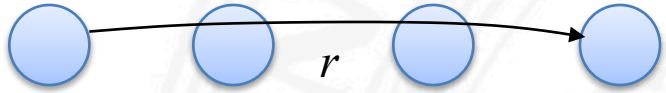
- unary relation: property
- n-ary relation: relation, constraint

Graphical representation

- binary relation:



- n-ary relation:



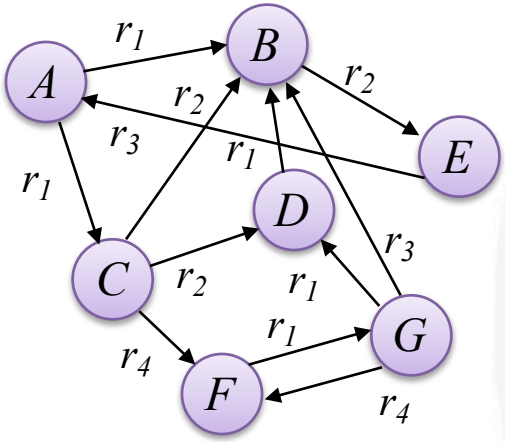
"hypergraph"

Object Recognition by Relational Matching

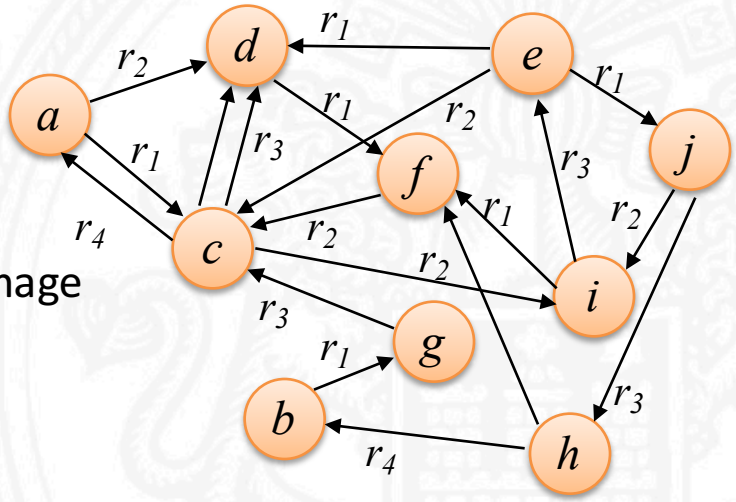
Principle:

- construct relational model(s) for object class(es)
- construct relational image description
- compute R-morphism (best partial match) between image and model(s)
- top-down verification with extended model

model



image



Compatibility of Relational Structures

Different from graphs, nodes and edges of relational structures may represent entities with rich distinctive descriptions.

Example: nodes = image regions with diverse properties

edges = spatial relations

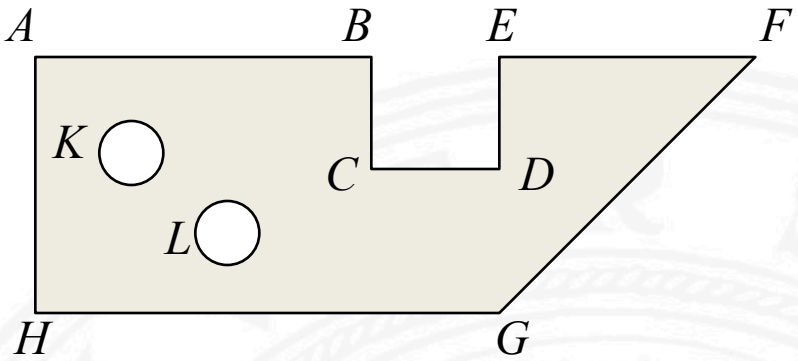
1. Compatibility of nodes
   An image node is compatible with a model node, if the properties of the nodes match.

2. Compatibility of edges
   An image edge is compatible with a model edge, if the edge types match.

3. Compatibility of structures
   A relational image description B is compatible with a relational model M, if there exists a bijective mapping of nodes of a partial structure B' of B onto nodes of a partial structure M' of M such that
   - corresponding nodes and edges are compatible
   - M is described by M' with sufficient completeness

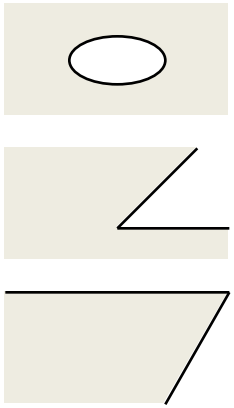
Example of a Relational Model I

Shape to be recognized:



Primitive descriptive elements (nodes)

properties



hole

t	type T_1
f	area
a	axes relation

interior corner

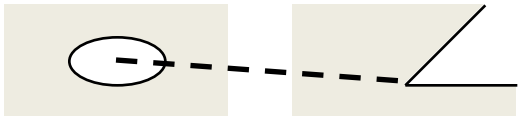
t	type T_2
w	angle

exterior corner

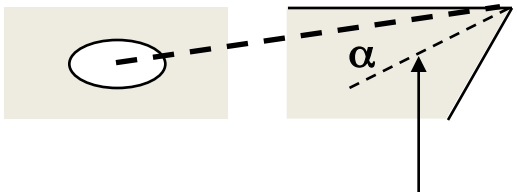
t	type T_3
w	angle

Example of a Relational Model II

Relations between primitive descriptive elements (edges):



...
d10 distance 10 ± 1
d12 distance 12 ± 1
d14 distance 14 ± 1
...



bisector of angle

...
o10 orientation 10 ± 5
o20 orientation 20 ± 5
o30 orientation 30 ± 5
...

Example of a Relational Model III

A	t	T_3
	w	90

E	t	T_3
	w	90

B	t	T_3
	w	90

F	t	T_3
	w	45

C	t	T_2
	w	90

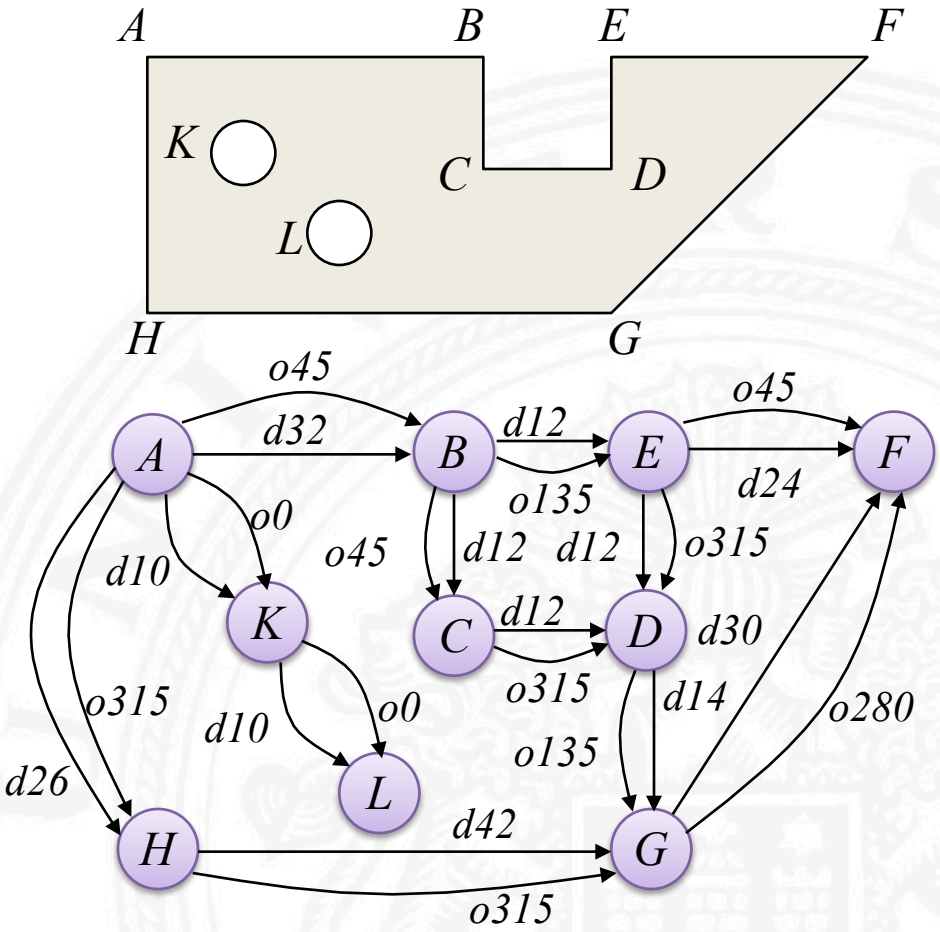
G	t	T_3
	w	135

D	t	T_2
	w	90

H	t	T_3
	w	90

K	t	T_1
	f	48
	a	1

L	t	T_1
	f	48
	a	1

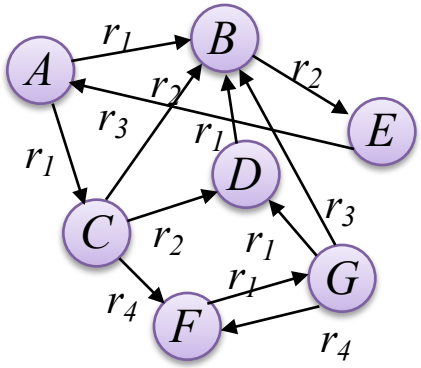


(not all edges are shown)

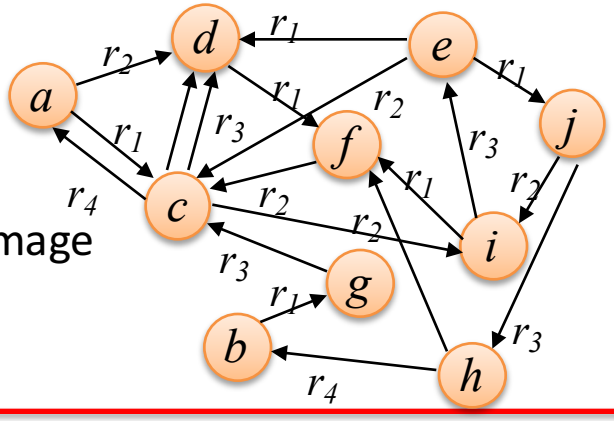
Relational Match Using a Compatibility Graph

nodes of compatibility graph = pairs with compatible properties
edges of compatibility graph = compatible pairs
cliques in compatibility graph = compatible partial structures

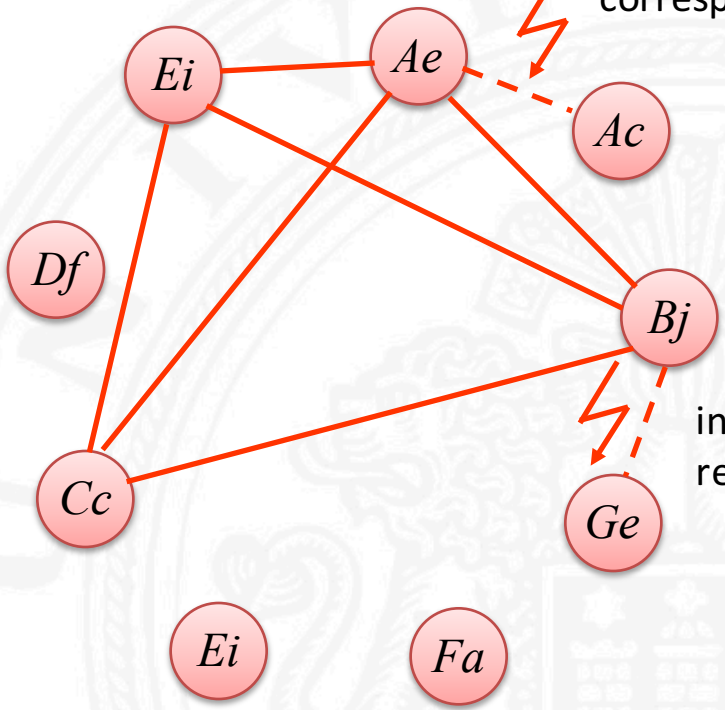
model



image



violates unique correspondence



incompatible relations

compatibility graph (not complete)

Finding Maximal Cliques

clique = complete subgraph

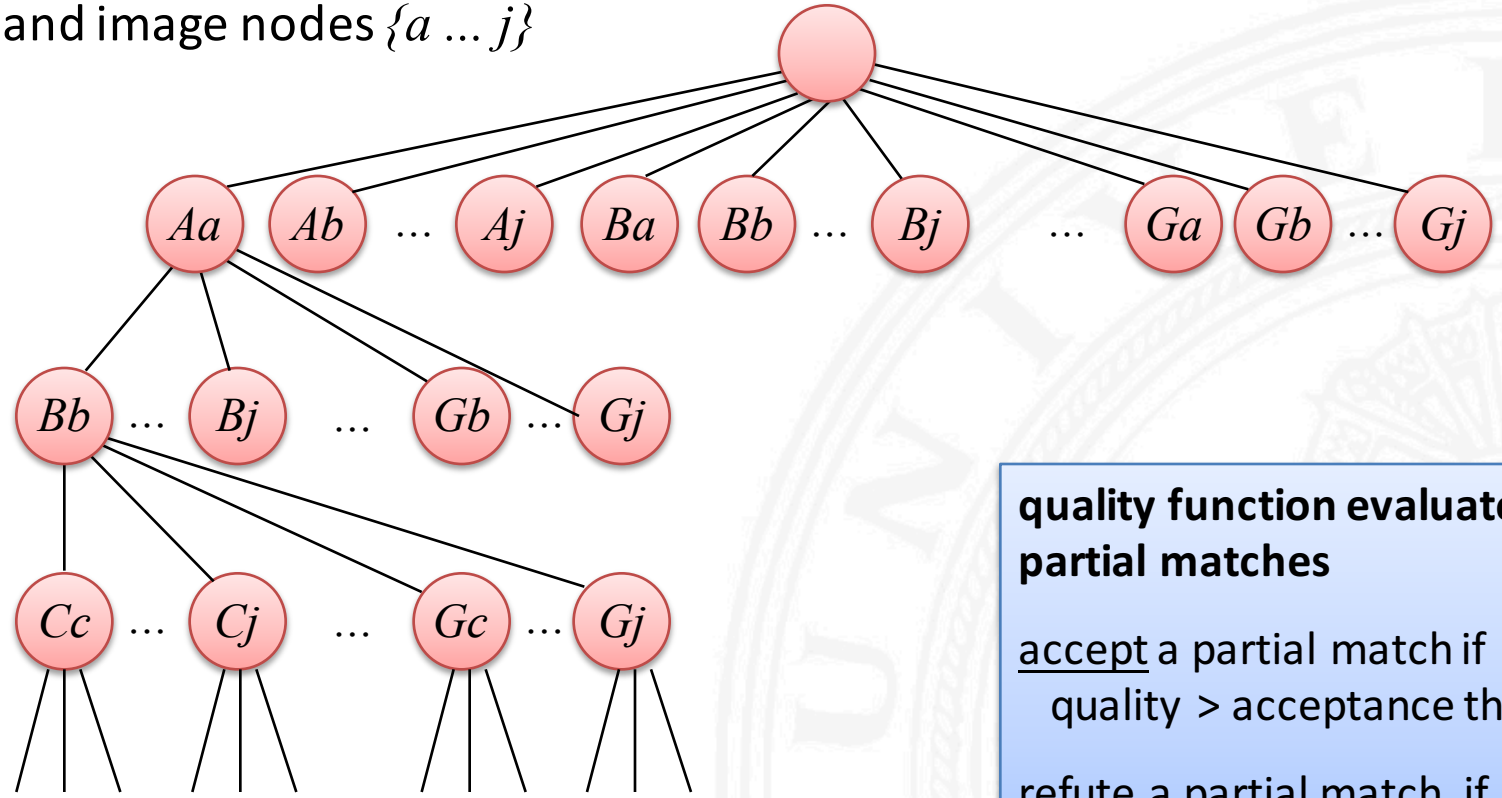
Find maximal cliques in a given compatibility graph

Many algorithms are available in the literature.

- Complexity is exponential relative to number of nodes of compatibility graph
- Efficient (suboptimal) solutions based on heuristic search

Relational Matching with Heuristic Search

Stepwise correspondence search between model nodes $\{A \ldots G\}$ and image nodes $\{a \ldots j\}$



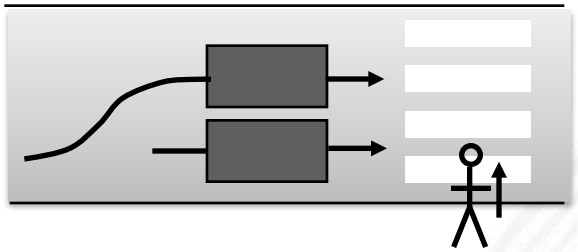
quality function evaluates partial matches

accept a partial match if
quality > acceptance threshold

refute a partial match, if
quality < refutation threshold

Example for Relational Event Recognition I

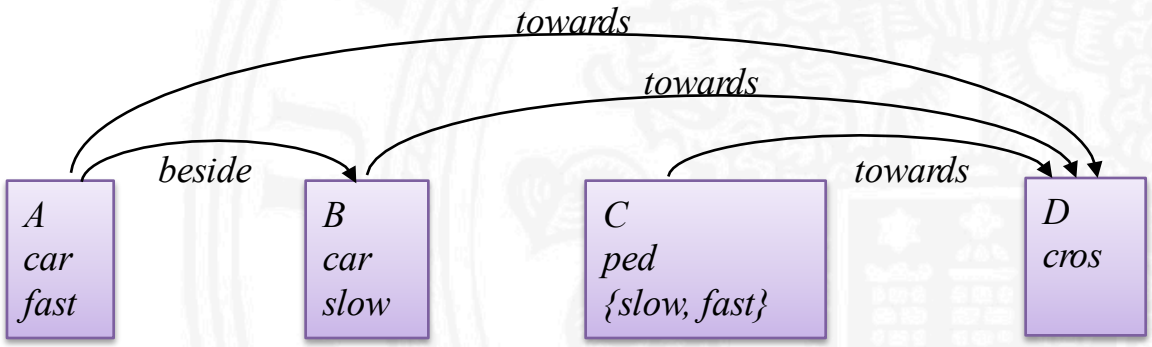
Suppose we want to recognise dangerous overtake events at pedestrian crossings:



Nodes: *type: {car, pedestrian, plain_road, crossing}*
speed: {zero, slow, fast}

Relations: *{beside, behind, before, towards, away}*

Model for dangerous overtake events:



(temporal relations not shown for simplicity)

Example for Relational Event Recognition II

Is there a dangerous overtake event in the following exciting traffic scene?

a car fast	*a behind b*
b car fast	*b before a*
c car slow	*b towards k*
d car slow	*c away k*
e ped fast	*d towards k*
f ped slow	*b beside d*
g ped slow	*f towards l*
h ped slow	*g towards k*
i ped slow	*e beside h*
k cros	*i away k*
l plain_road	*h towards l*

- Apply heuristic search!
- What heuristic may be useful?
- How can the approach be improved?