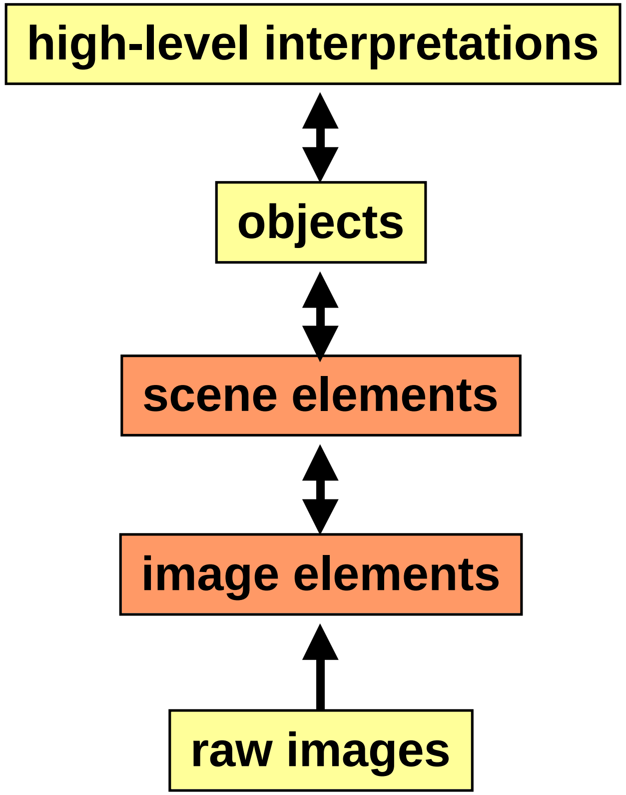
General Principles of 3D Image Analysis



Extraction of 3D information from an image (sequence) is important for

- vision in general (= scene reconstruction)
- many tasks (e.g. robot grasping and navigation, traffic analysis)
- not all tasks (e.g. image retrieval, quality control, monitoring)

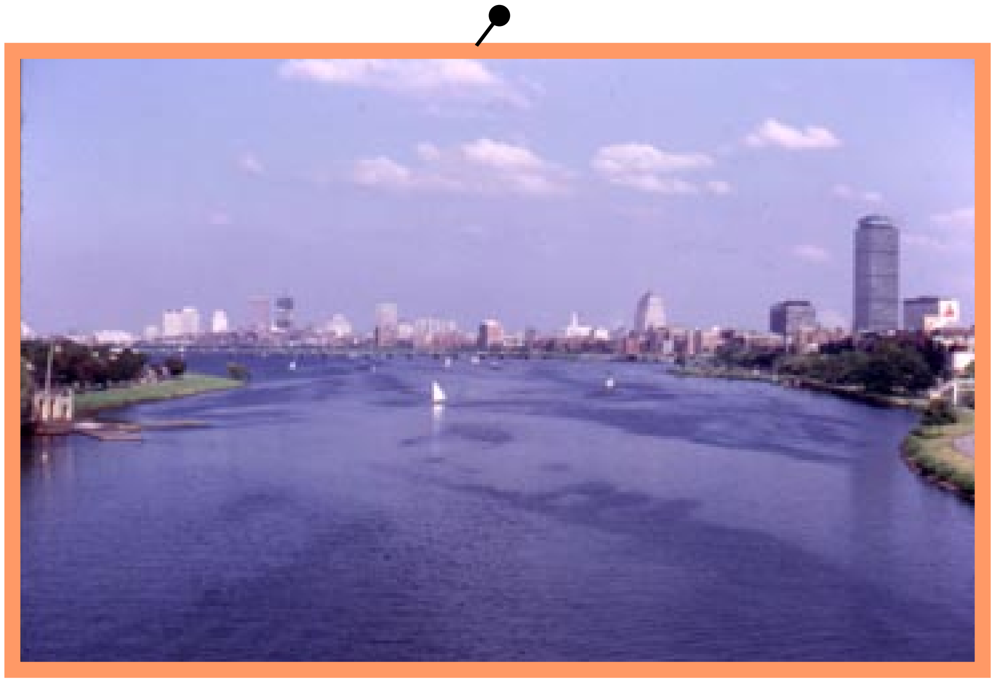
Recovery of 3D information is possible

- by multiple cameras (e.g. binocular stereo)
- by a monocular image sequence with motion + weak assumptions
- by a single image + strong assumptions or prior knowledge about the scene

Single Image 3D Analysis

Humans exploit various cues for a tentative (heuristic) depth analysis:

- **size of known objects**
- **texture gradient**
- **occlusion**
- **colour intensities**
- **angle of observation**
- **continuity assumption**
- **generality assumption**



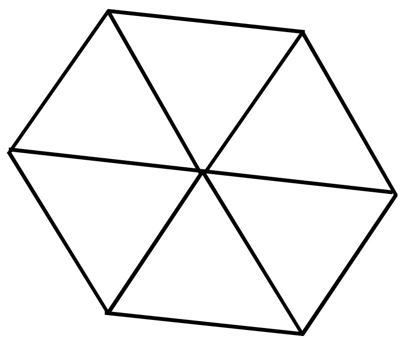
Generality Assumption

Assume that

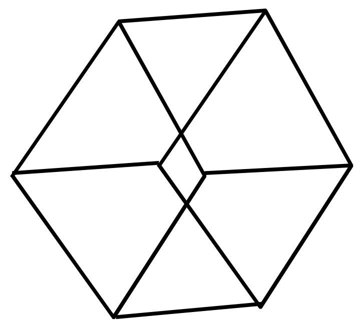
- viewpoint
- illumination
- physical surface properties

are general, i.e. do not produce coincidental structures in the image.

<u>Example</u>: Do not interpret this figure as a 3D wireframe cube, because this view is not general.



General view:



The generality assumption is the basis for several specialized interpretation methods, e.g.

- shape from texture
- shape from shading

...

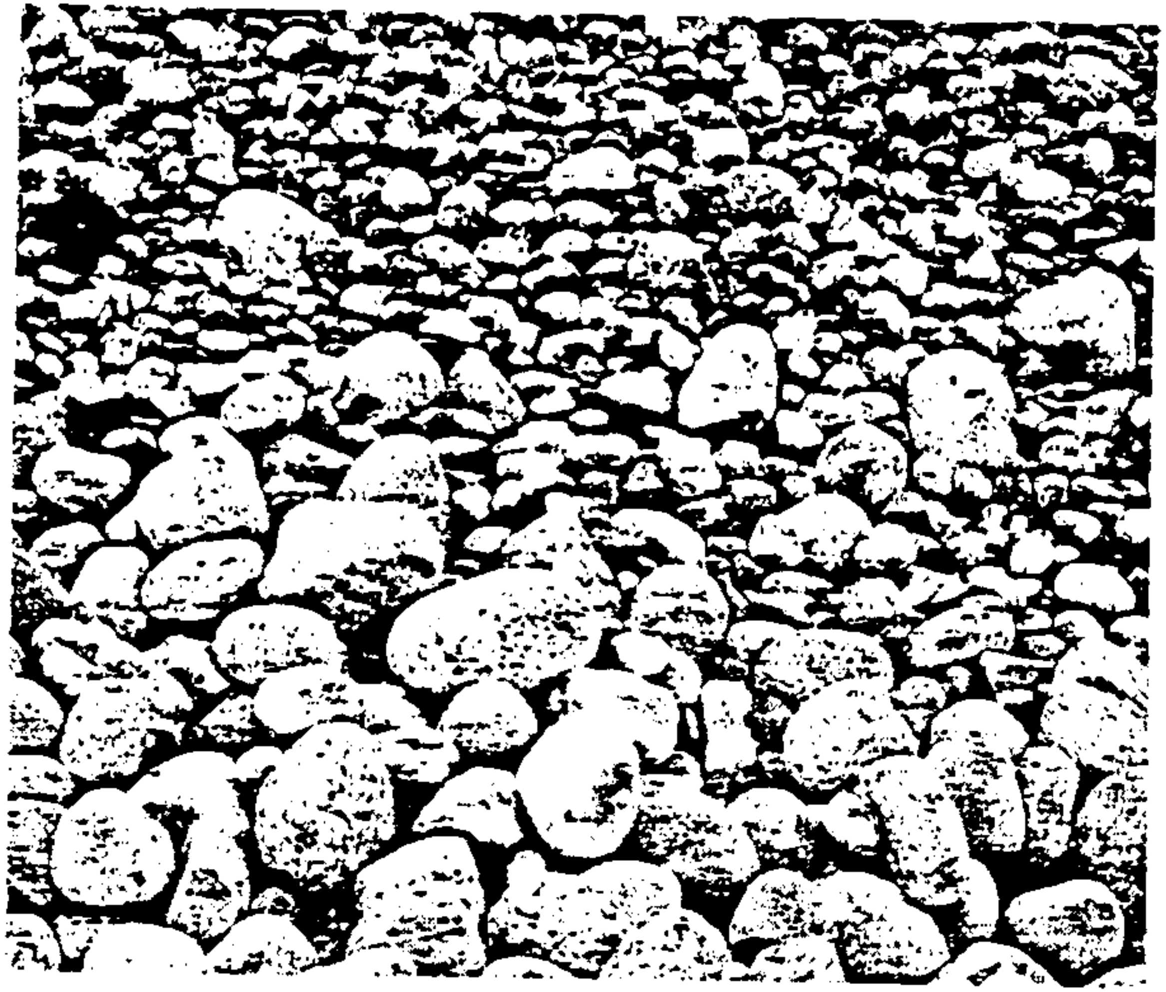
- "shape from X"

Texture Gradient

Assume that texture does not mimick projective effects

Interpret texture gradient as a 3D projection effect

(Witkin 81)

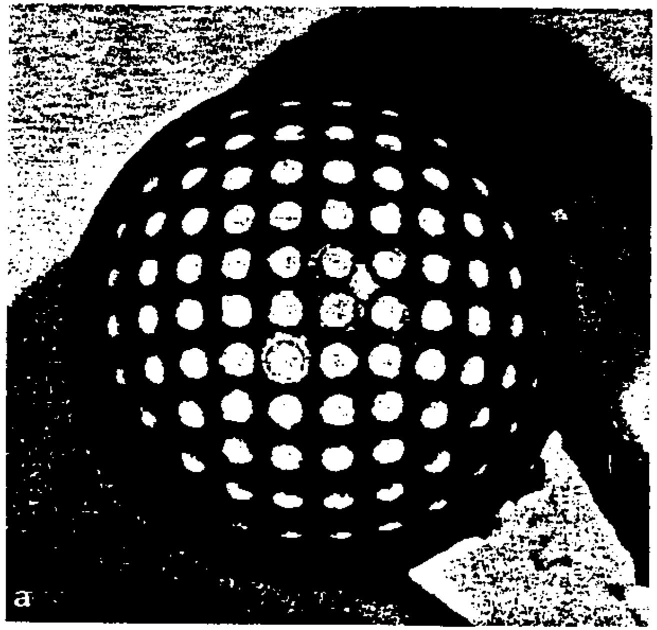


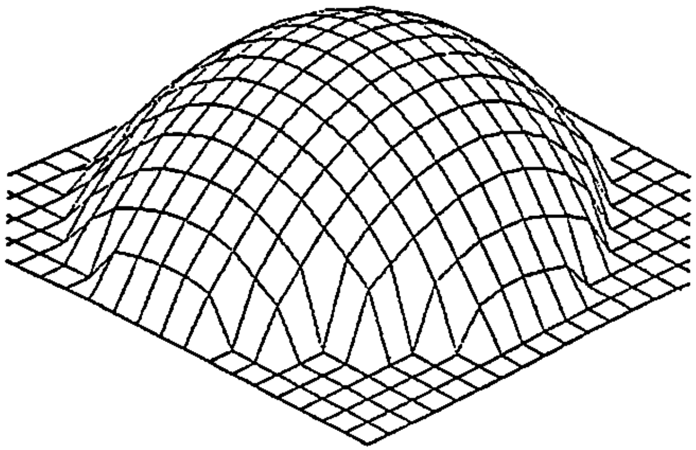
Shape from Texture

Assume
- homogeneous texture on 3D surface and
- 3D surface continuity

Reconstruct 3D shape from perspective texture variations

(Barrow and Tenenbaum 81)



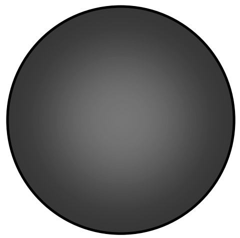


Surface Shape from Contour

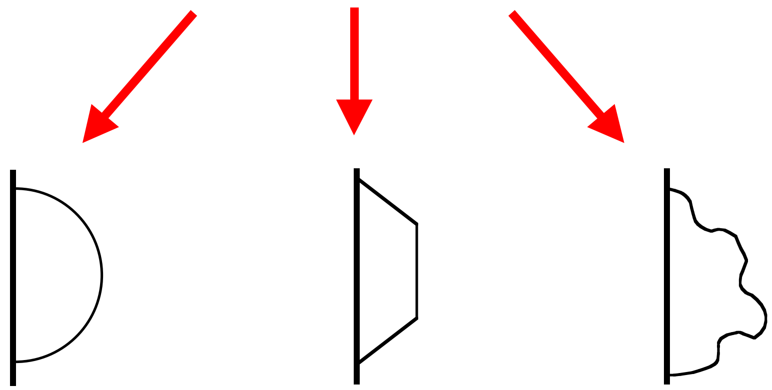
Assume "non-special" illumination and surface properties

3D surface shape maximizes probability of observed contours and minimizes probability of additional contours

2D image contour



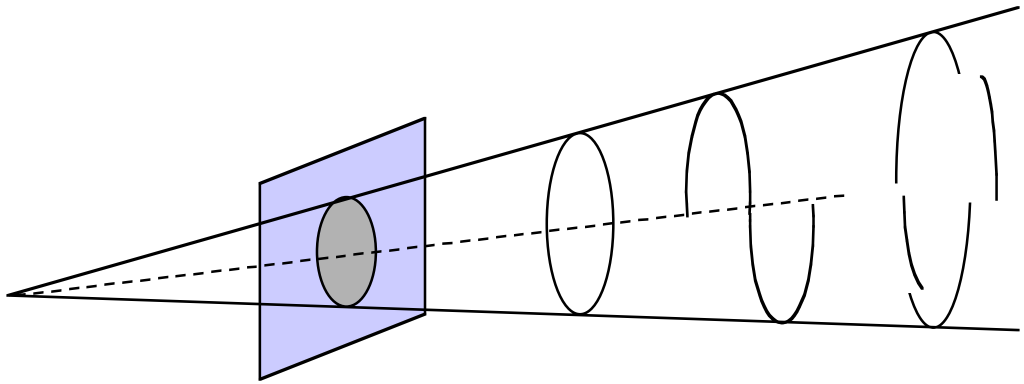
possible 3D reconstructions



a **b** **c**

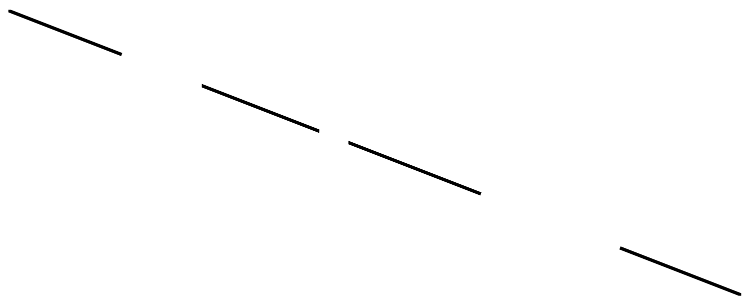
3D Line Shape from 2D Projections

Assume that lines connected in 2D are also connected in 3D



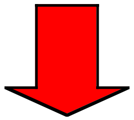
Reconstruct 3D line shape by minimizing spatial curvature and torsion

2D collinear lines are also 3D collinear

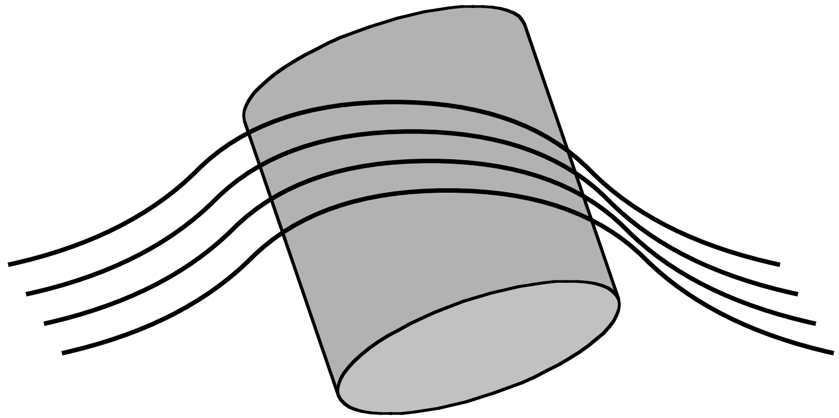


3D Shape from Multiple Lines

Assume that similar line shapes result from similar surface shapes



Parallel lines lie locally on a cylinder

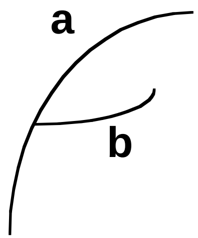


(Stevens 81)

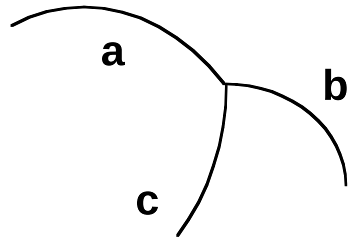
3D Junction Interpretation

rules for junctions
of curved lines

(Binford 81)



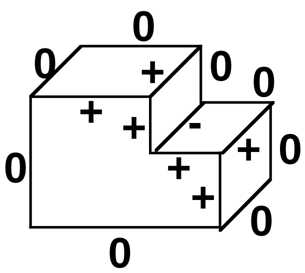
a not behind b



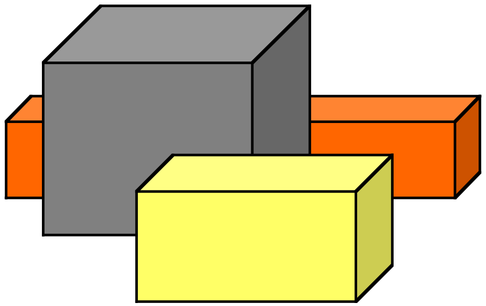
a, b and c meet

rules for blocks-
world junctions

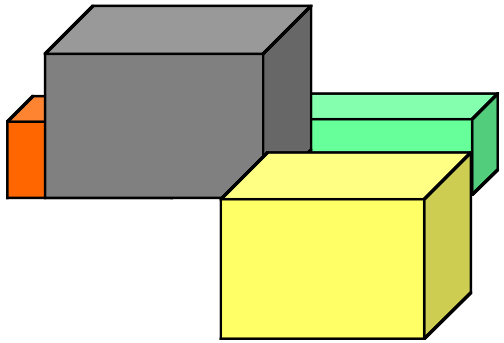
(Waltz 86)



edge labels



"general" ensemble



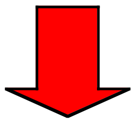
"special" ensemble

3D Line Orientation from Vanishing Points

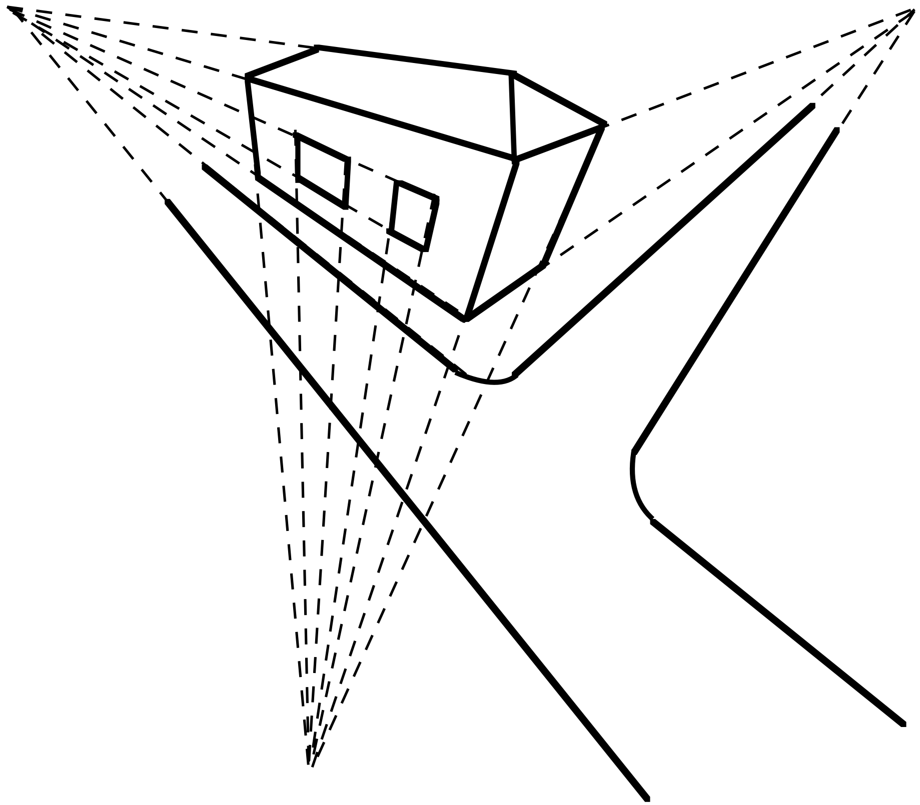
From the laws of perspective projection:

The projections of 3D parallel straight lines intersect in a single point, the vanishing point.

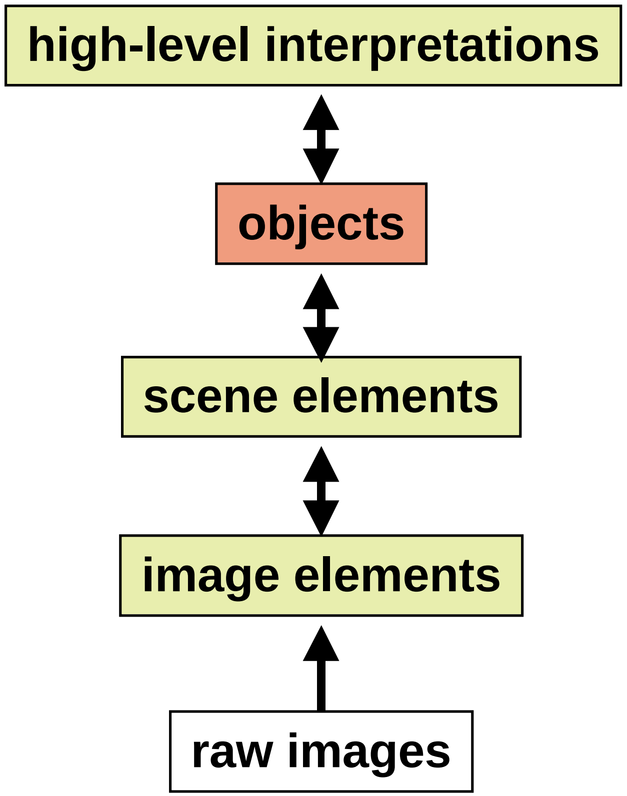
Assume that more than 2 straight lines do not intersect in a single point by coincidence



If more than 2 straight lines intersect, assume that they are parallel in 3D



Object Recognition



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

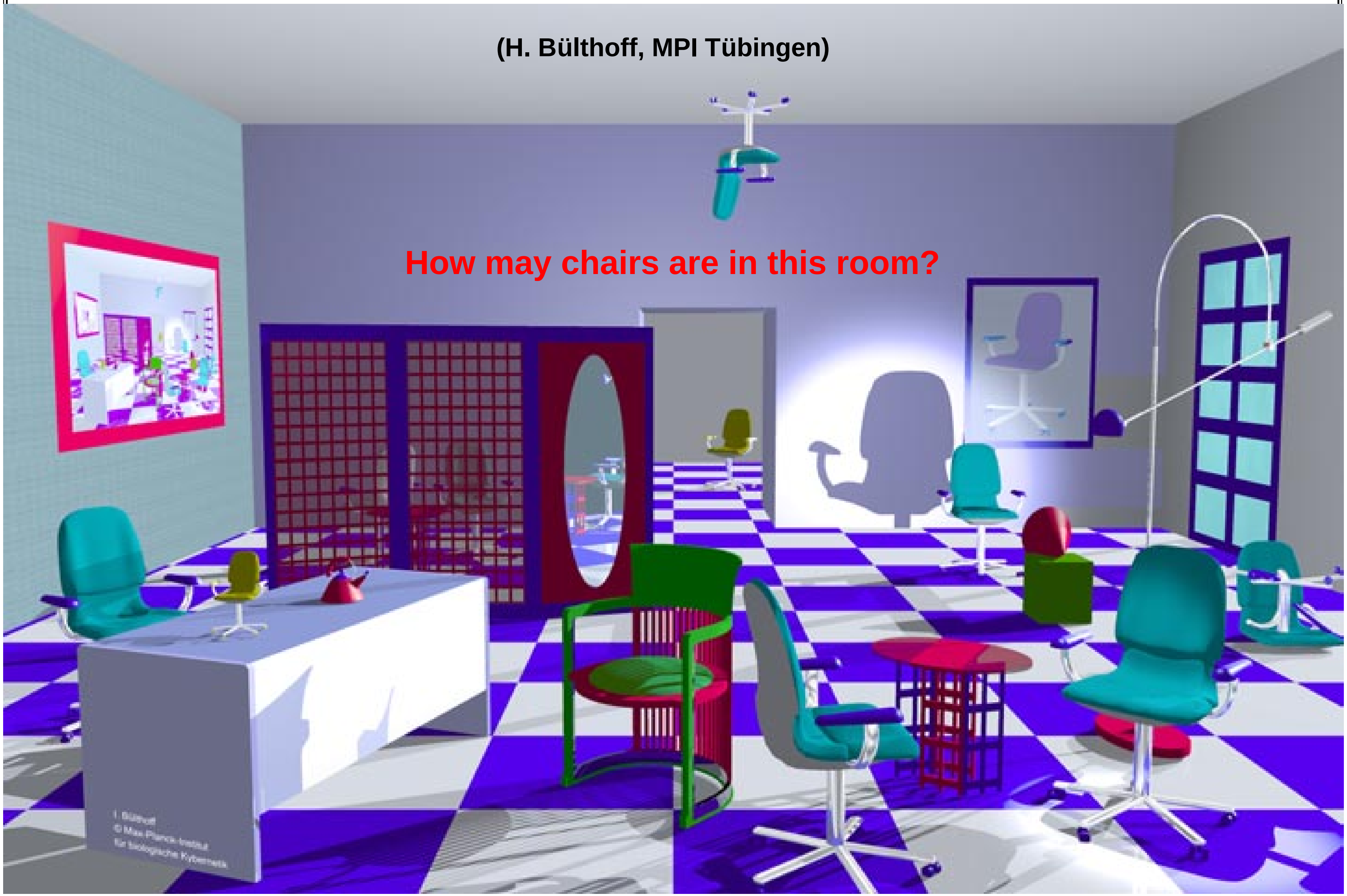
- object classification

assigning an object to one of a set of predetermined classes

- object categorization

assigning an object to an object category of biological vision

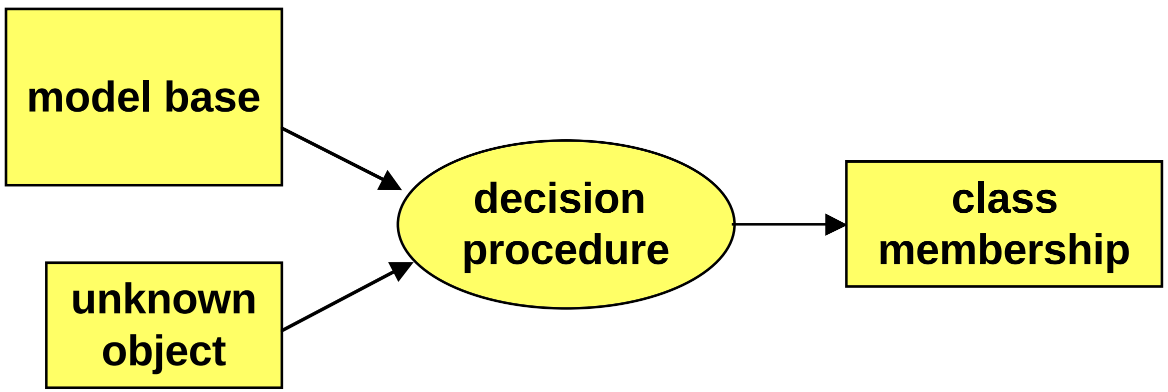
The Chair Room



About Model-based Recognition

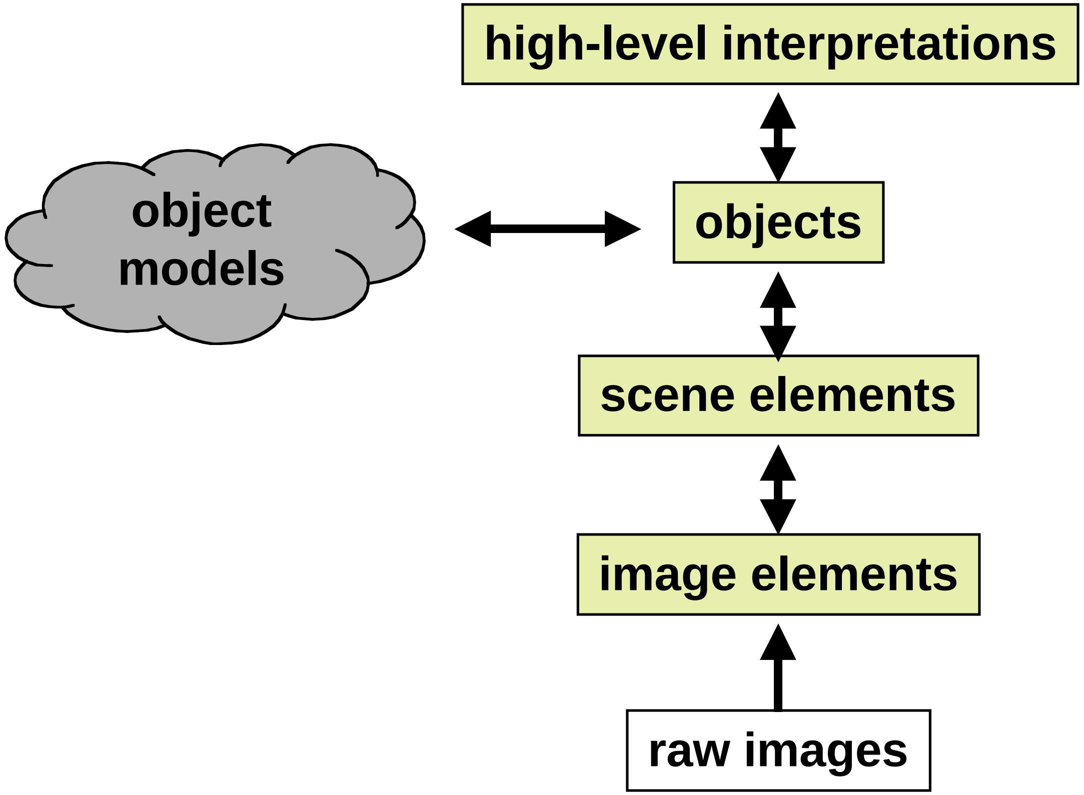
"model" = generic description of a class of objects

- explicit representation of object properties (as opposed to decision procedures which incorporate class properties implicitely)
- generic (class-independent) decision procedure
- reusable and incremental model bases
- no strict correspondence with biological vision



Model-based Object Recognition

How to classify objects based on a generic description.



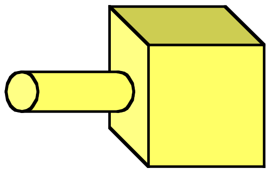
3D Models vs. 2D Models

1. Requirement:

Object models must represent invariant class properties

=> 3D models, properties independent of views

e.g.



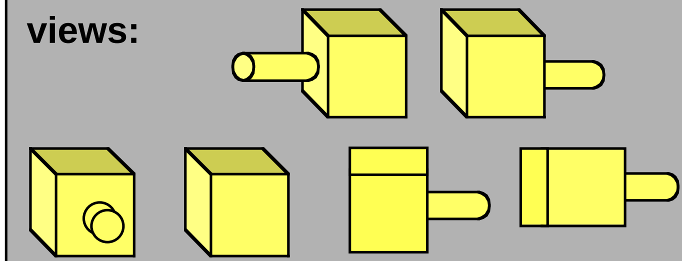
class hammer
is-a aggregat
has-parts part1, part2
is-a part1 cube
is-a part2 cylinder
coaxially-connected part1 part2

2. Requirement:

Object models must support recognition

=> 2D models, view-dependent properties

class hammer
views:

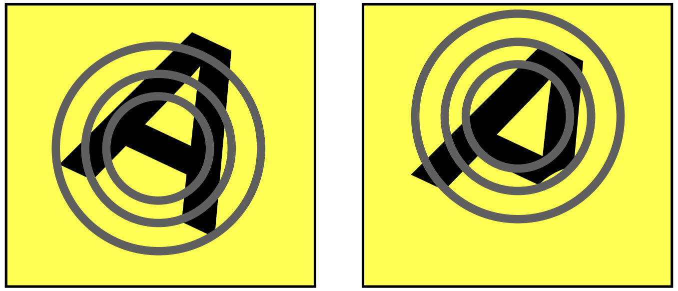


Holistic Models vs. Component Models

Holistic ("global") models:

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

z.B. 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:

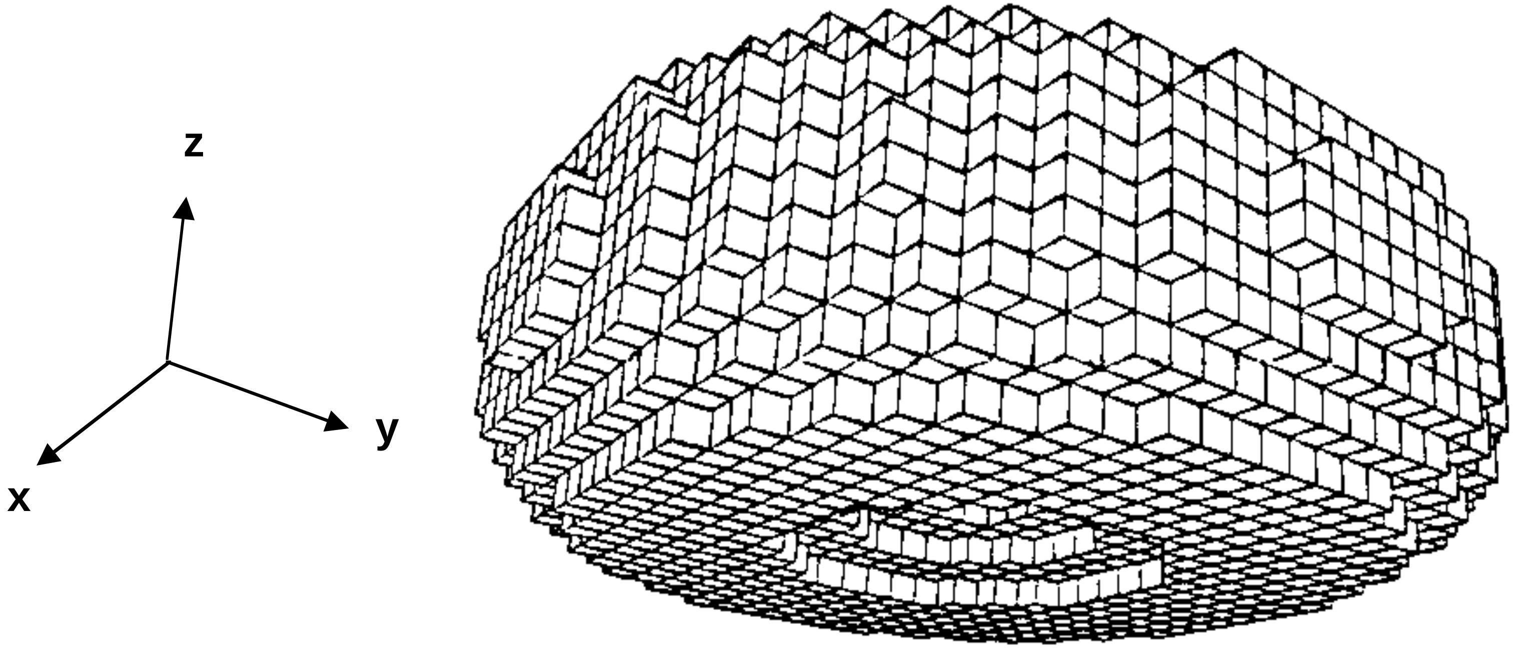
- **3D space occupancy**
- **Oct-trees**
- **CSG ("Constructive Solid Geometry") models**
- **3D surface triangulation**

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

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

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

3D Space Occupancy Model

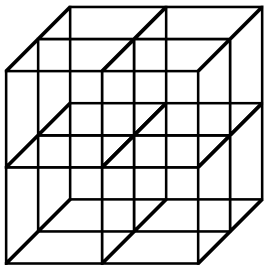


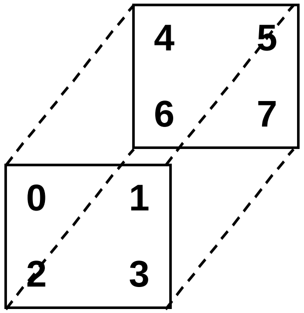
3D shape represented by cube primitives

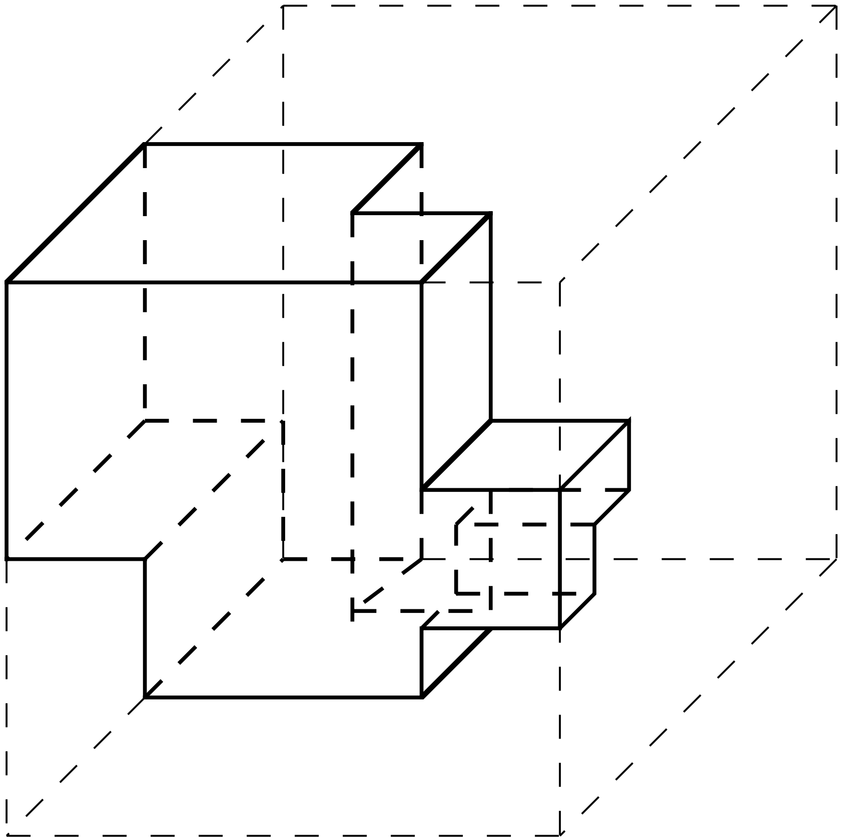
- **useful for highly irregular shapes (e.g. medical domain)**
- **useful for robotics applications (e.g. collision avoidance)**
- **interior cubes do not provide information relevant for views**
- **no explicit surface properties (e.g. surface normals)**

Oct-trees

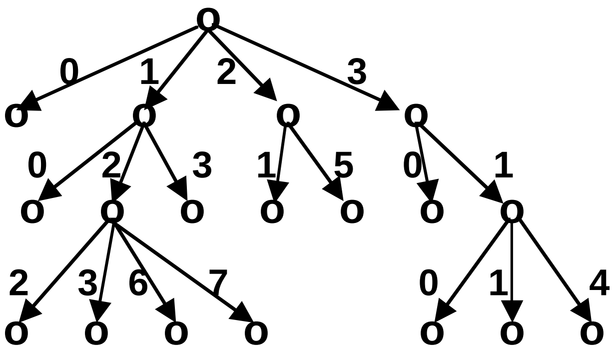
- hierarchical 3D shape model
- analog to 2D quad-trees
- each cube is recursively decomposed into 8 subcubes
- access via numbering code





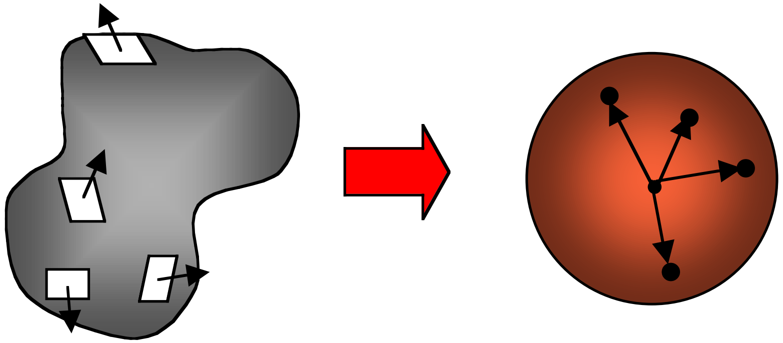


Oct-tree for example (right):



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 $\{(o_1\ a_1)\ (o_2\ a_2)\ ...\ (o_N\ a_N)\}$
- **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**

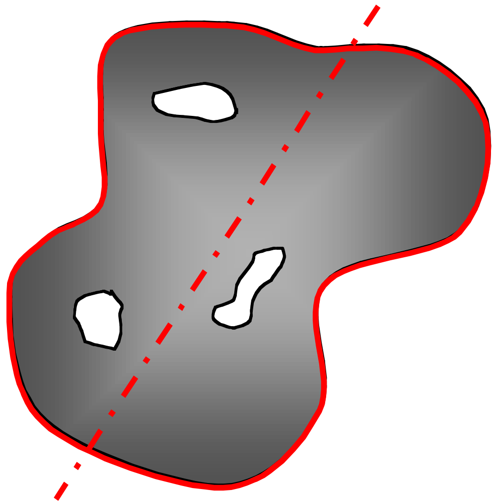
Illustration of EGI Recognition Procedure

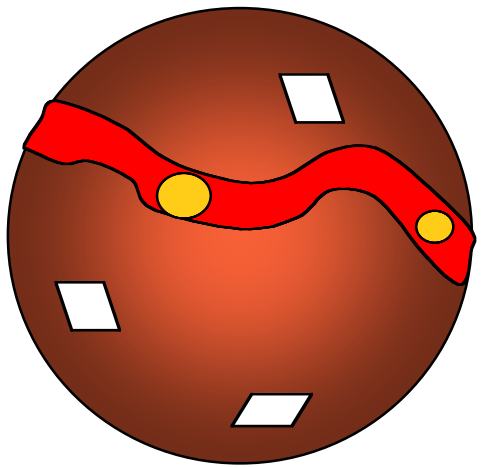
1. **Determine direction of axis of minimal inertia**
   => locations on EGI with corresponding entries

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





Representing Axial Bodies

Picasso´s "Rites of Spring" shows bodies composed of roughly cylindrical and cone-shaped pieces.

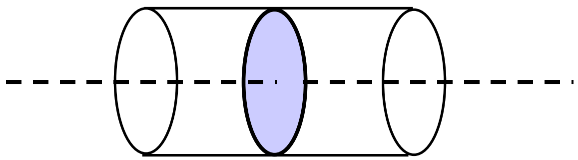
What representations can capture the inherent restrictions of such shapes?

(a)

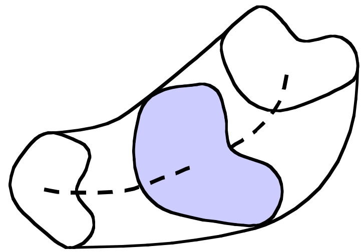
Generalized Cylinders

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

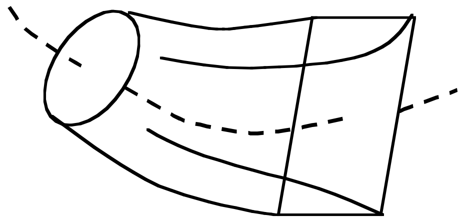
ordinary <u>cylinder</u> swept out by a circle along a straight line



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



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

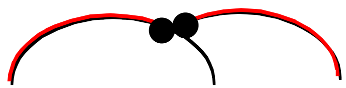


Generalized cones were used in ACRONYM (Brooks et al. 79) to model mainly artificial objects, e.g. airplanes. Under certain conditions, the 3D surface may be reconstructed from the contours of many views.

Conditions for 3D Reconstruction from Contours

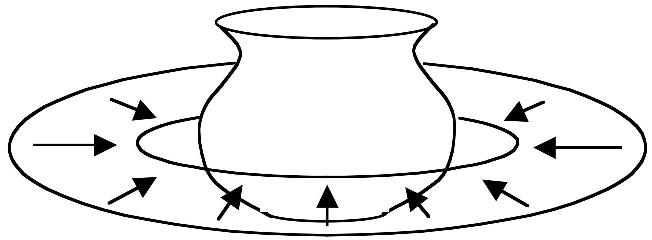
1. **Each line of sight touches the body at a single point**
   => we see a "contour generator"
2. **Nearby points on the contour in the image are also nearby in 3D (with only few exceptions)**

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)



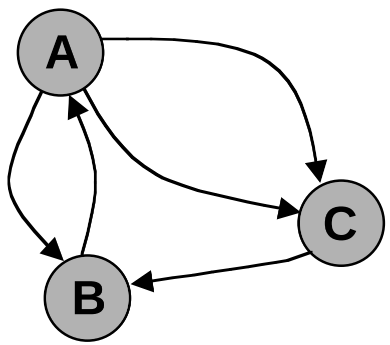
Relational Models

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

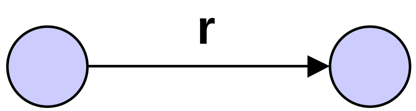
Relations between Components

unary relation: **property**

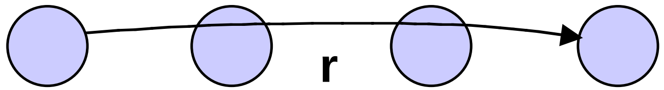
n-ary relation: **relation, constraint**

<u>Graphical representation</u>

binary relation:



n-ary relation:



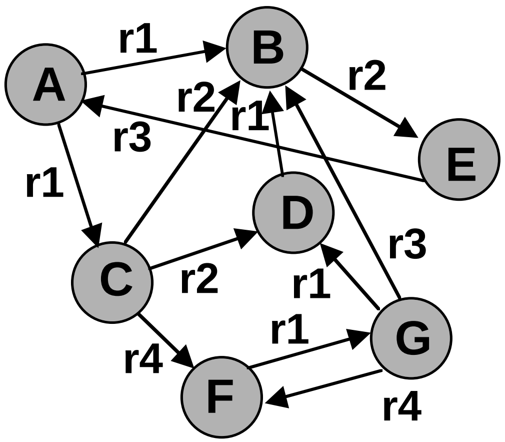
"hypergraph"

Object Recognition by Relational Matching

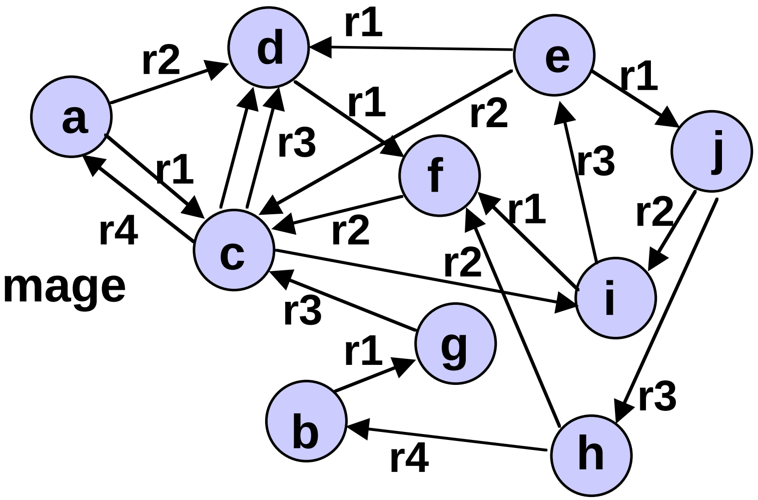
<u>Principle</u>:

- **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.

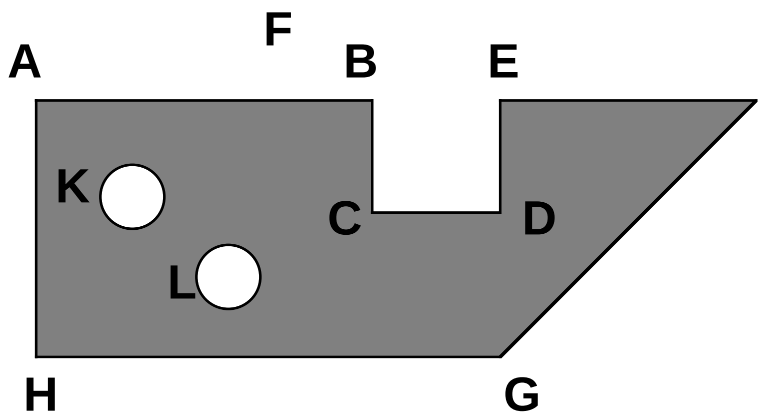
3. Compatibility of structures

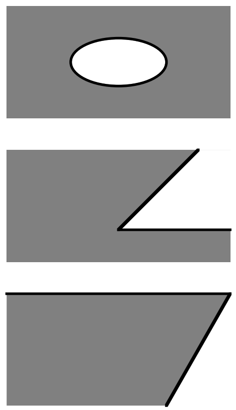
A relational image description B is compatible with a relational model M, if there exists a bijective mapping of nodes of a partial structure B´of B onto nodes of a partial structure M´of M such that

- corresponding nodes and edges are compatible
- M is described by M´ with sufficient completeness

Example of a Relational Model (1)

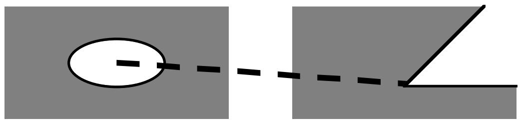
shape to be recognized:



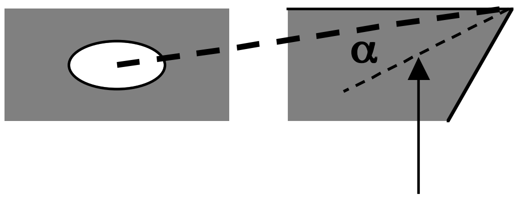
primitive descriptive elements (nodes)		properties	
	hole	t	type T1
		f	area
		a	axes relation
	interior corner	t	type T2
		w	angle
	exterior corner	t	type T3
		w	angle

Example of a Relational Model (2)

relations between primitive descriptive elements (edges)



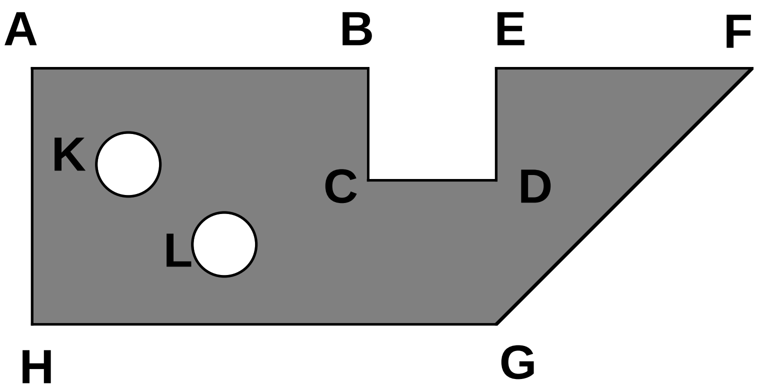
...	
d10	distance 10 ± 1
d12	distance 12 ± 1
d14	distance 14 ± 1
...	



bisector of angle

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

Example of a Relational Model (3)



A	t	T3
	w	90

B	t	T3
	w	90

C	t	T2
	w	90

D	t	T2
	w	90

E	t	T3
	w	90

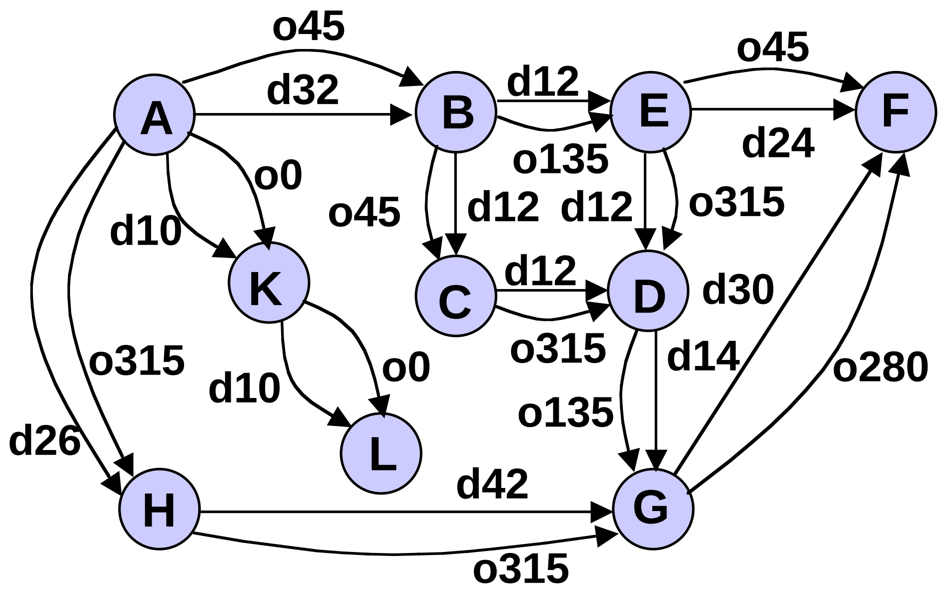
F	t	T3
	w	45

G	t	T3
	w	135

H	t	T3
	w	90

K	t	T1
	f	48
	a	1

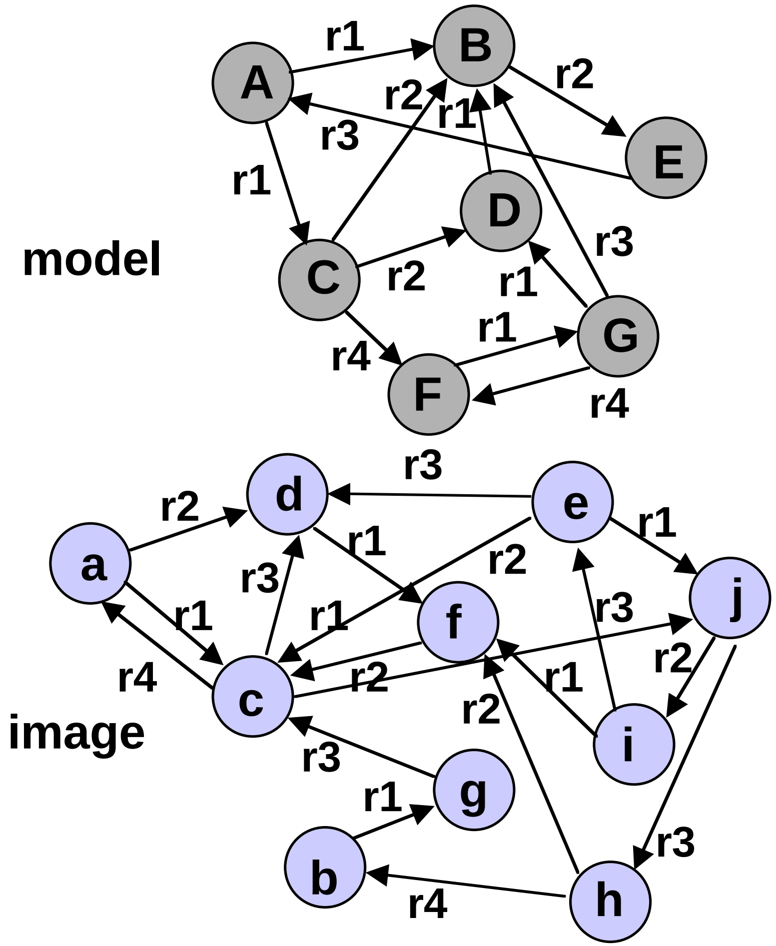
K	t	T1
	f	48
	a	1

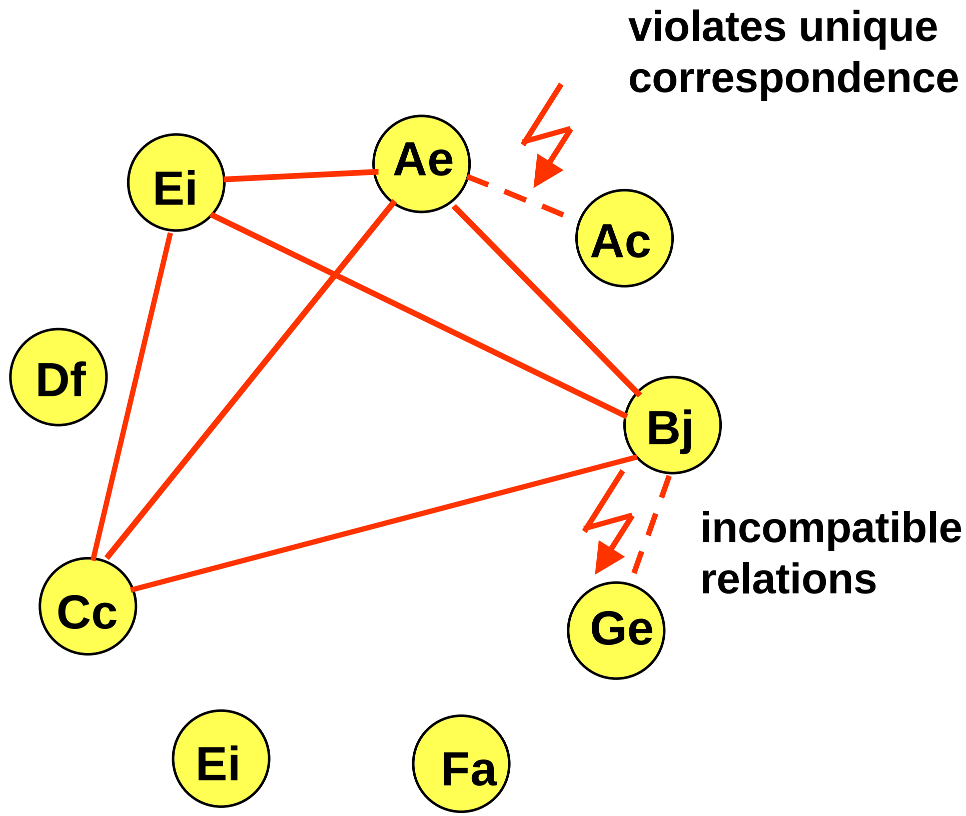


(not all edges are shown)

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





compatibility graph (not complete)

Finding Maximal Cliques

clique = complete subgraph

Find maximal cliques in a given compatibility graph

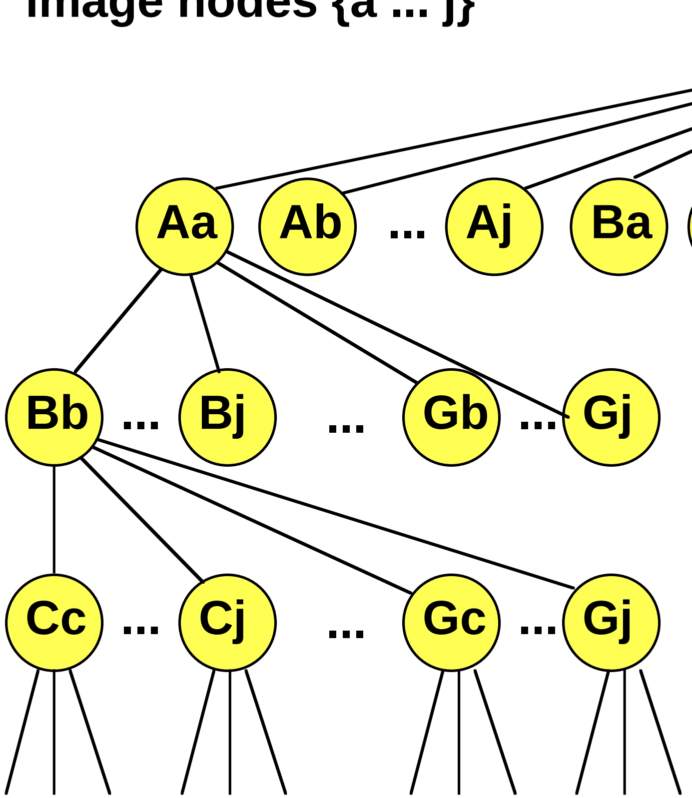
Algorithms are available in the literature, e.g.

Bron & Kerbusch, Finding all Cliques of an Undirected Graph, Communications of the ACM, Vol. 16, Nr. 9, S. 575 - 577, 1973.

- **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 ... G} and image nodes {a ... j}



- **quality function evaluates partial matches**
- **accept a partial match if quality > acceptance threshold**
- **refute a partial match, if quality < refutation threshold**

Optimization Techniques

Search for an <u>optimal</u> interpretation of an image in terms of object models requires the optimization of a function of merit (quality, confidence) or <u>objective function</u>.

Objective functions may be defined in many ways:

- **distance or similarity measure between models and instances**
- **likelihoods or posterior probabilities**
- **information content**
- **utility measure**

There are many ways to optimize an objective function, prominently:

- **heuristic search (e.g. best-first, hill-climbing, A*)**
- **constraint solving**
- **relaxation**
- **simulated annealing**
- **genetic algorithms**

Danger: Solution procedures may deliver a local optimum instead of the global optimum.

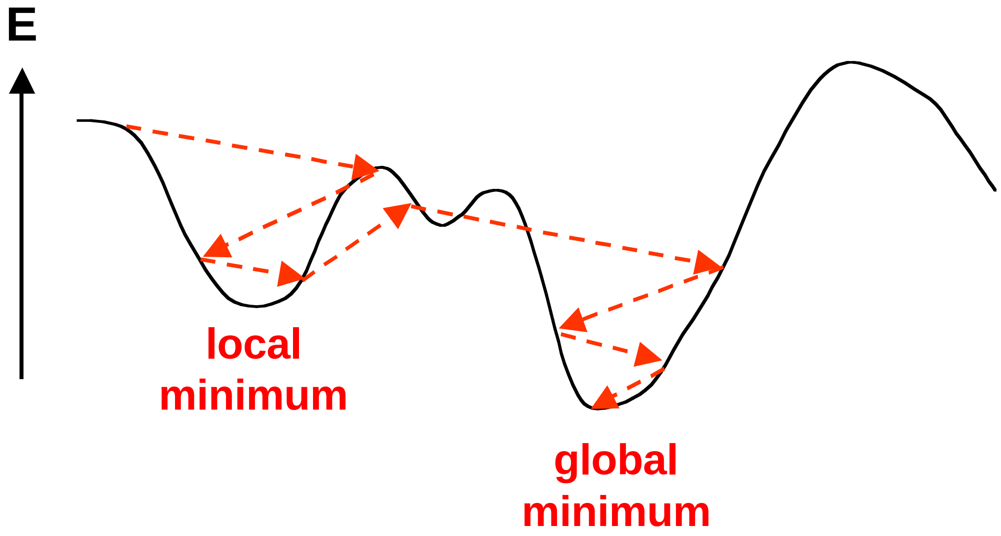
Simulated Annealing (1)

Simulated annealing ("simuliertes Tempern") is an optimization process in analogy to energy minimization in atomic crystalization processes.

In this context: optimum = minimum of objective function

Main ideas:

- **mix of down-hill steps (toward minimum) and up-hill steps (to ecape local minima)**
- **"cooling" process slowly reduces up-hill activities**



finding the global minimum by random exploration

Simulated Annealing (2)

Physical model calls for random steps with resulting energy change ΔE.

$\Delta E \le 0$ execute step

$\Delta E > 0$ execute step with probability $P(\Delta E) = \exp(\frac{-\Delta E}{k_B T})$

Simulated annealing algorithm:

$\underline{x}$ is vector of optimization parameters, $J(\underline{x})$ is objective function

1. **Choose initial value $\underline{x}$, compute $J(\underline{x})$**
2. **Perturb the parameter $\underline{x}$ slightly ceating $\underline{x}'$ and compute $J(\underline{x}')$**
3. **Generate a random number r from a uniform distribution in the interval (0, 1).**

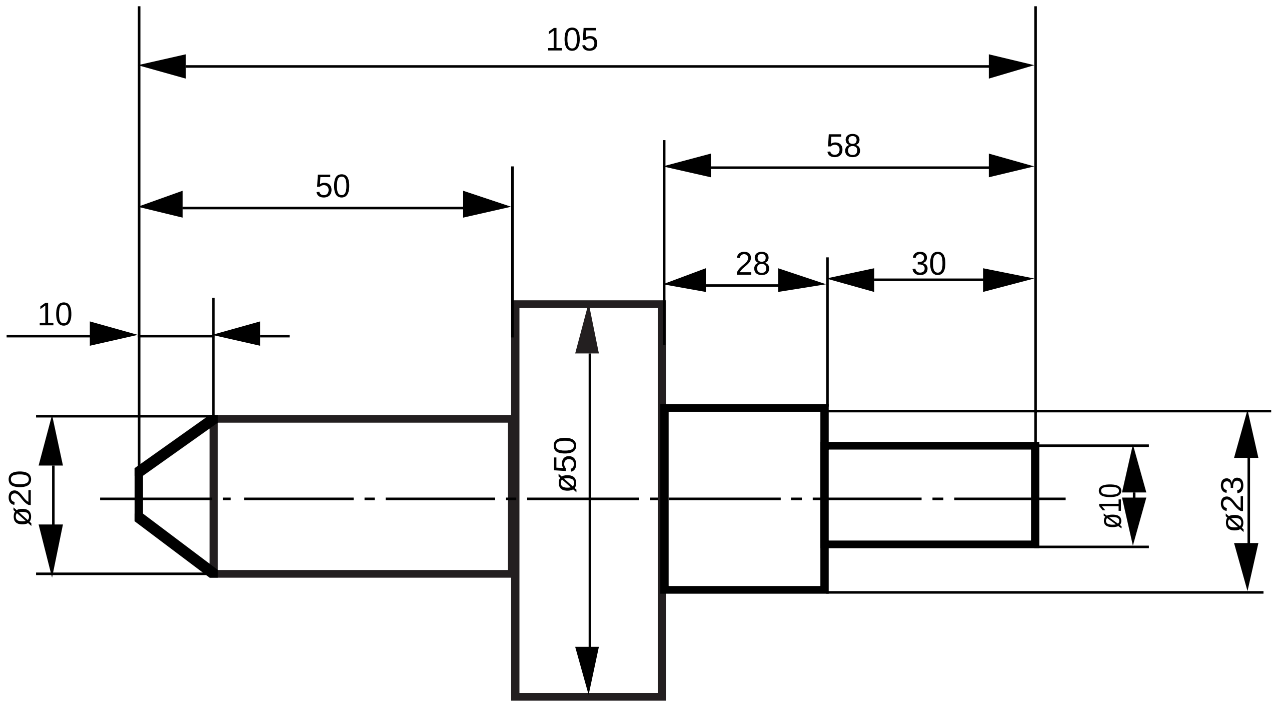
   If $r < \exp\left\{\frac{-(J(\underline{x}') - J(\underline{x}))}{k_B T}\right\}$

   then assign $\underline{x} = \underline{x}'$ and $J(\underline{x}) = J(\underline{x}')$.
4. **Repeat steps 2 and 3 n(T) times for each temperature T.**
5. **Decrease temperature T according to annealing schedule. Repeat steps 2 to 4.**
6. **The resulting parameter vector $\underline{x}$ is the solution of the optimization problem.**

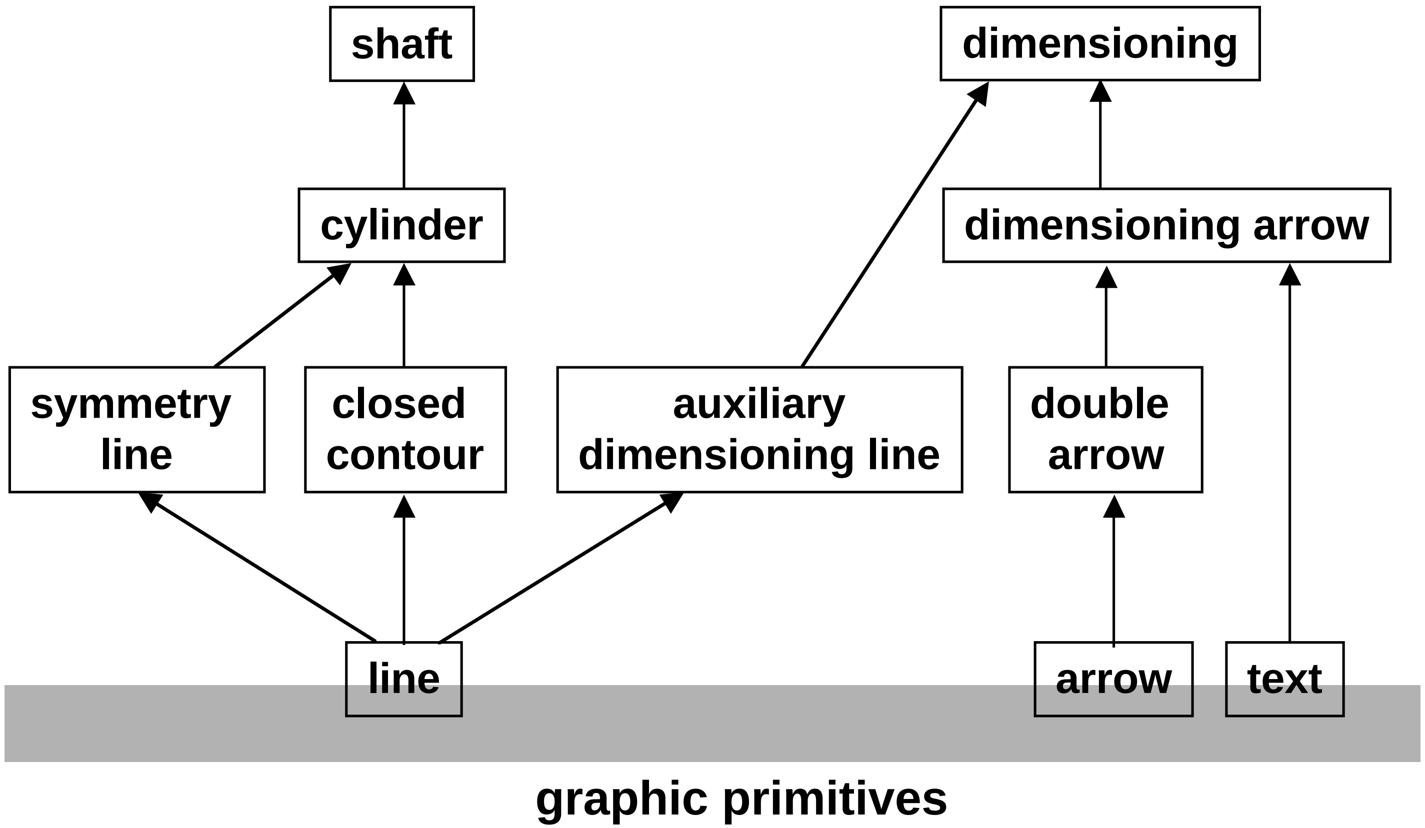
Case study: Drawing Interpretation

Transforming paper drawings into CAD formats (Pasternak 94)

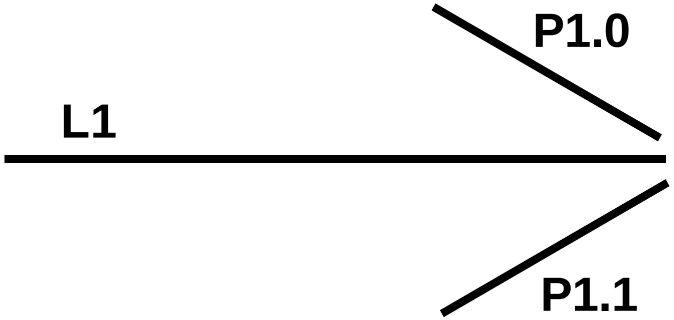
=> recognition of contours, dimensioning, symmetry lines, surface markings etc.



Partonomy of Object Parts

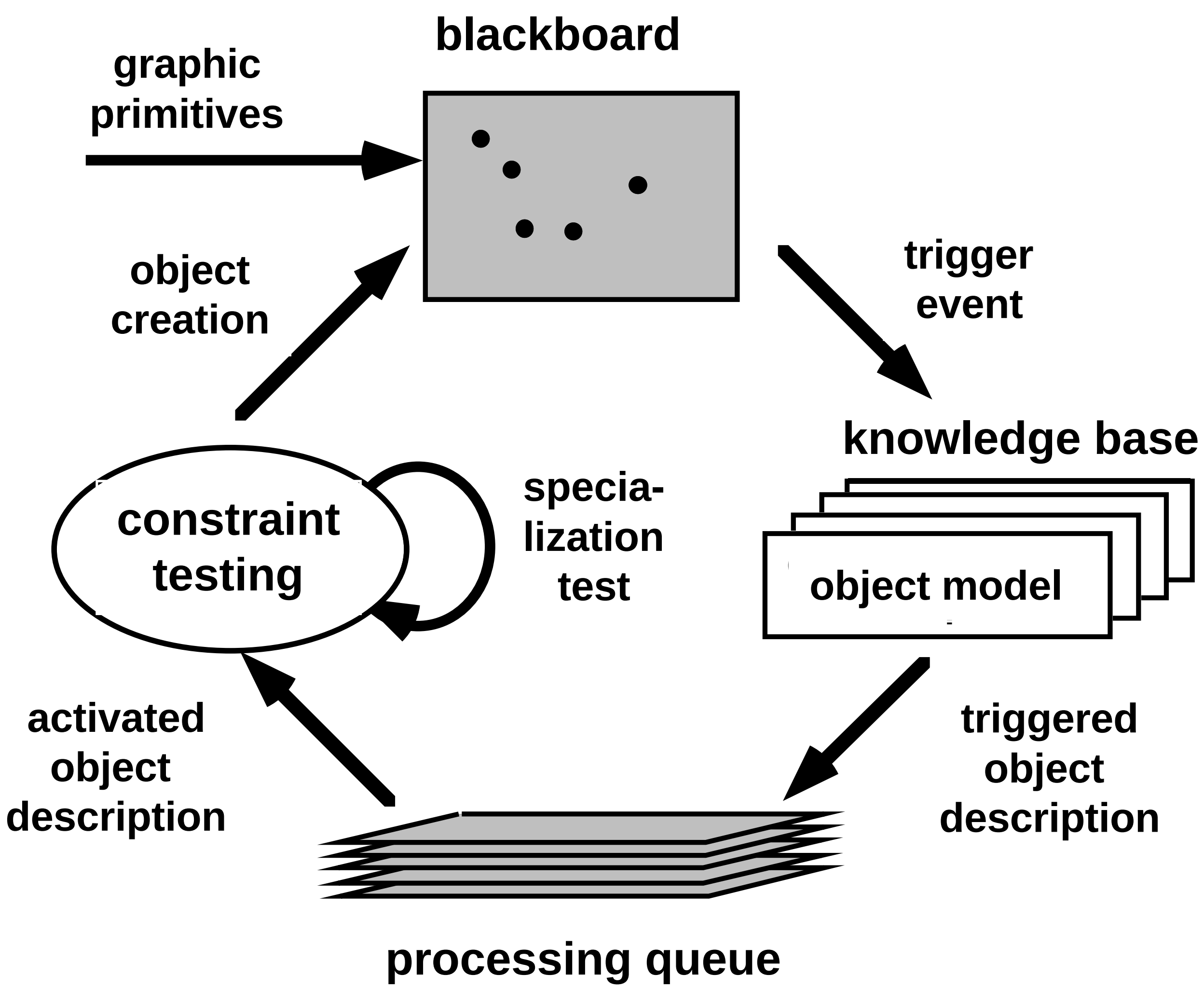


Specification of an Arrow



NAME:	arrow
KIND-OF:	symbol
PARTS:	L1 TYPE line, P1 TYPE polygon
TRIGGER:	P1
CONSTRAINTS:	NOT PART L1 P1 NEAR P1.0.end L1.start ANGLE P1.0.end L1.start [5 30] => ang NEAR P1.1.start L1.start ANGLE P1.10.start L1.start ang

Processing Cycle

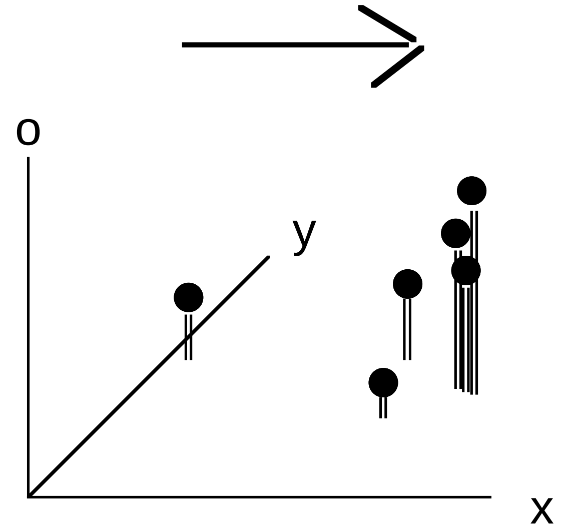


Property Spaces

Representation of graphical objects in multi-dimensional property spaces to allow effective object retrieval and access via their properties

Example:

arrow in 3D property space with endpoint coordinates x, y und orientation o

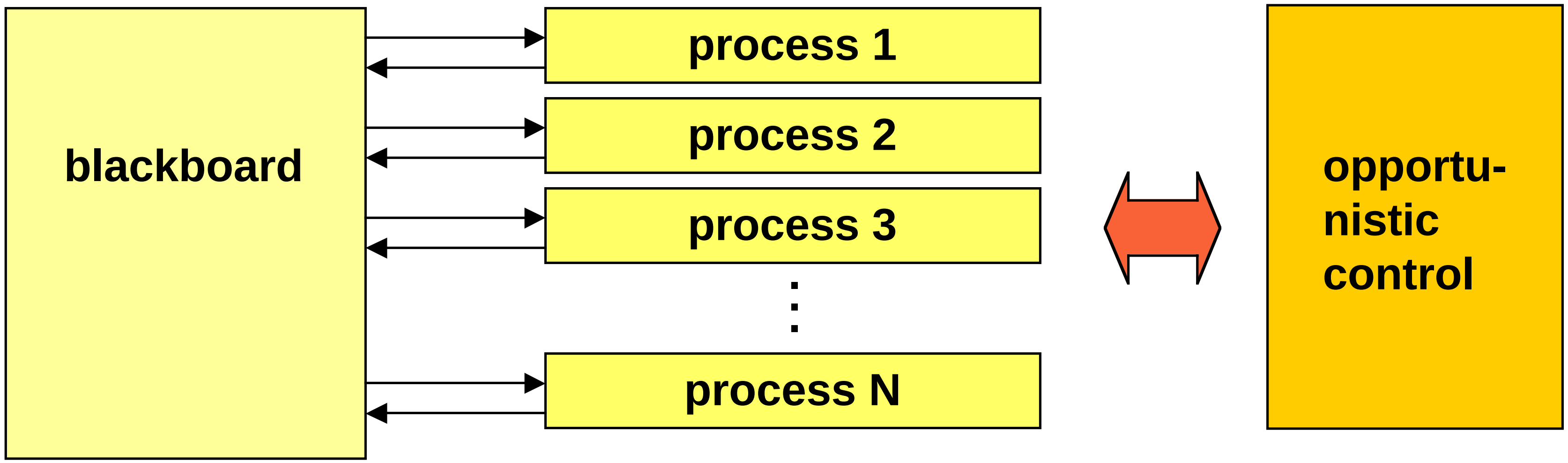


How to construct property spaces:

- **discretization (coarse quantization) of property values**
- **set-type property space cells to accommodate multiple objects with identical properties**
- **overlapping value ranges to avoid boundary effects**

Blackboard Architecture

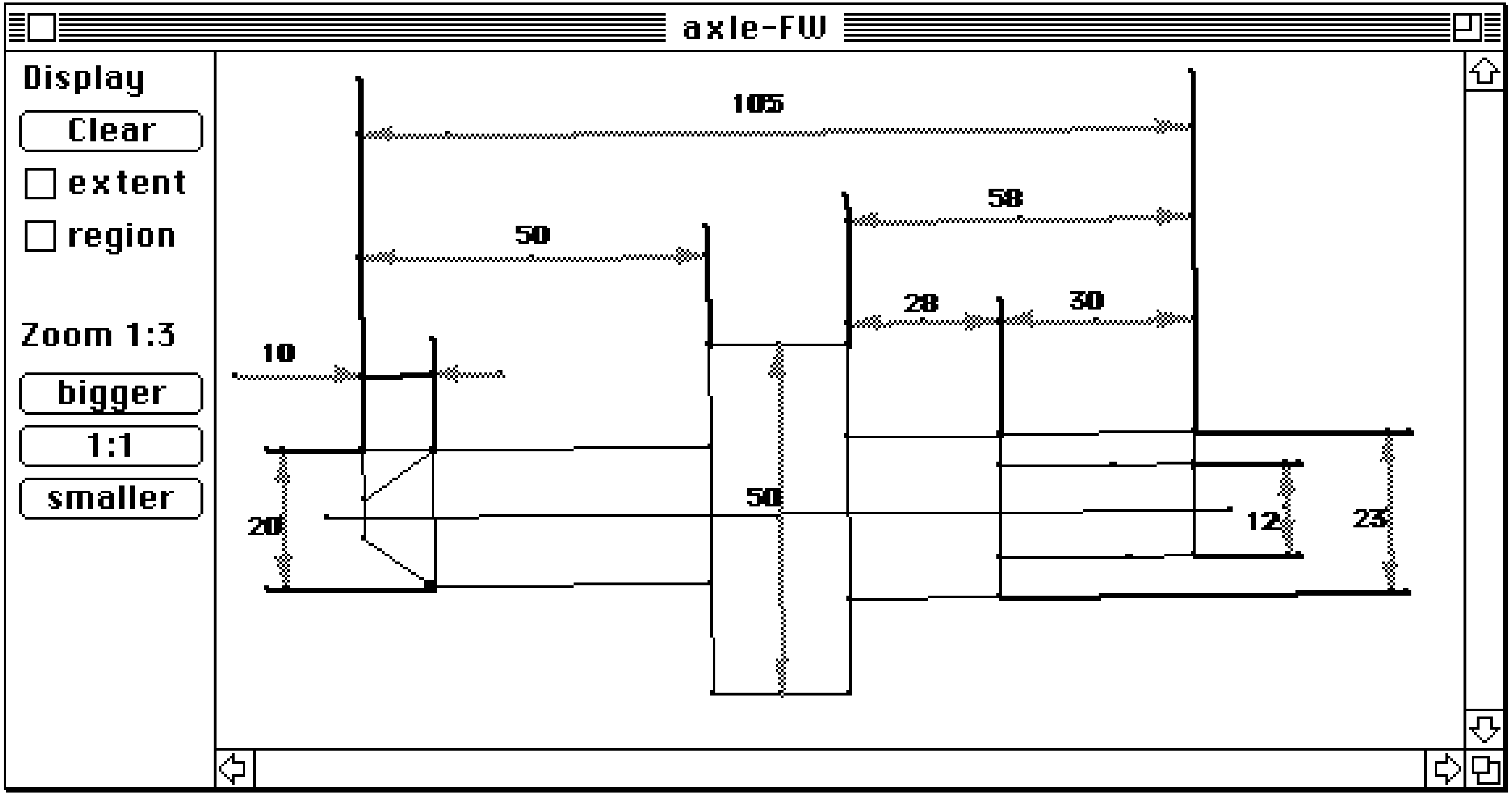
Independent processes communicate Prozesse via a common database ("blackboard")



Recognition process may be structured into processes dedicated to the recognition of individual components

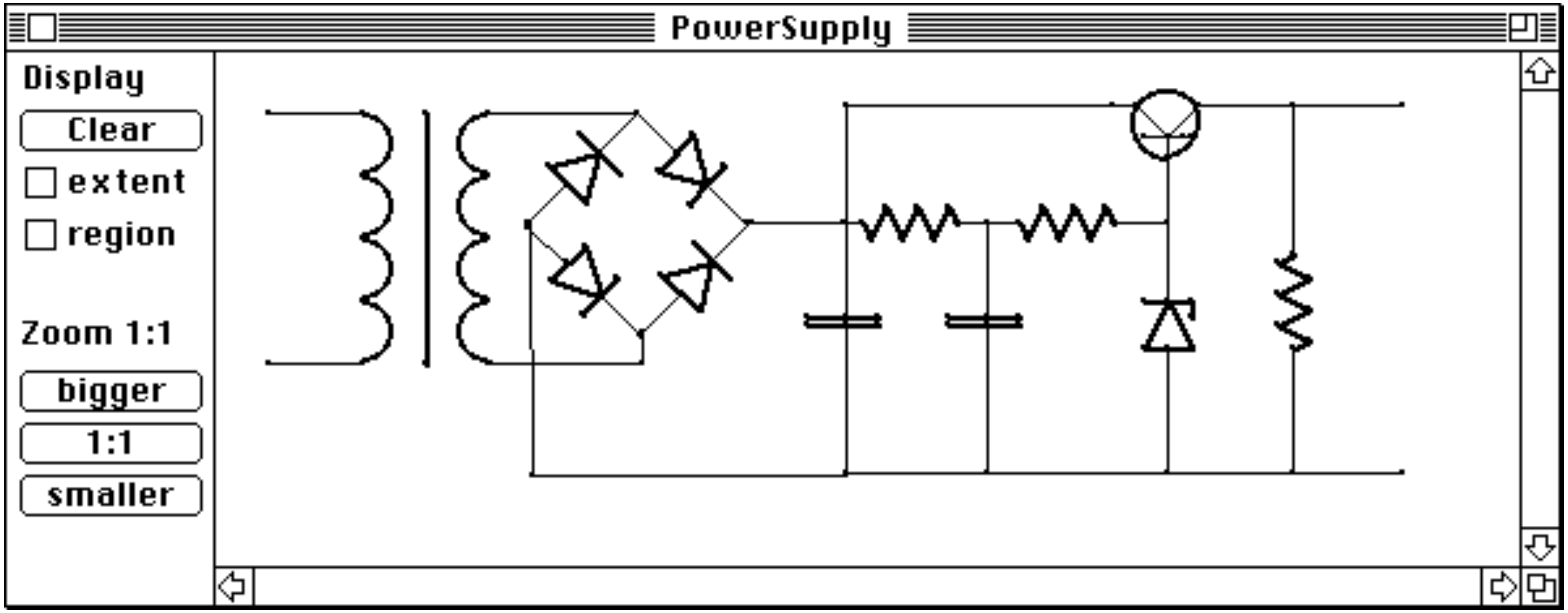
Analysis of a Machine Drawing

Recognition of dimensioning



Analysis of an Electrical Circuit

Recognition of electrical components



Qualitative Relations

Quantitative relations are characterized by a quantitative value, e.g.

$$D \subseteq O \times O \times R^{+}$$

with O = set of objects, R^{+} positive real numbers.

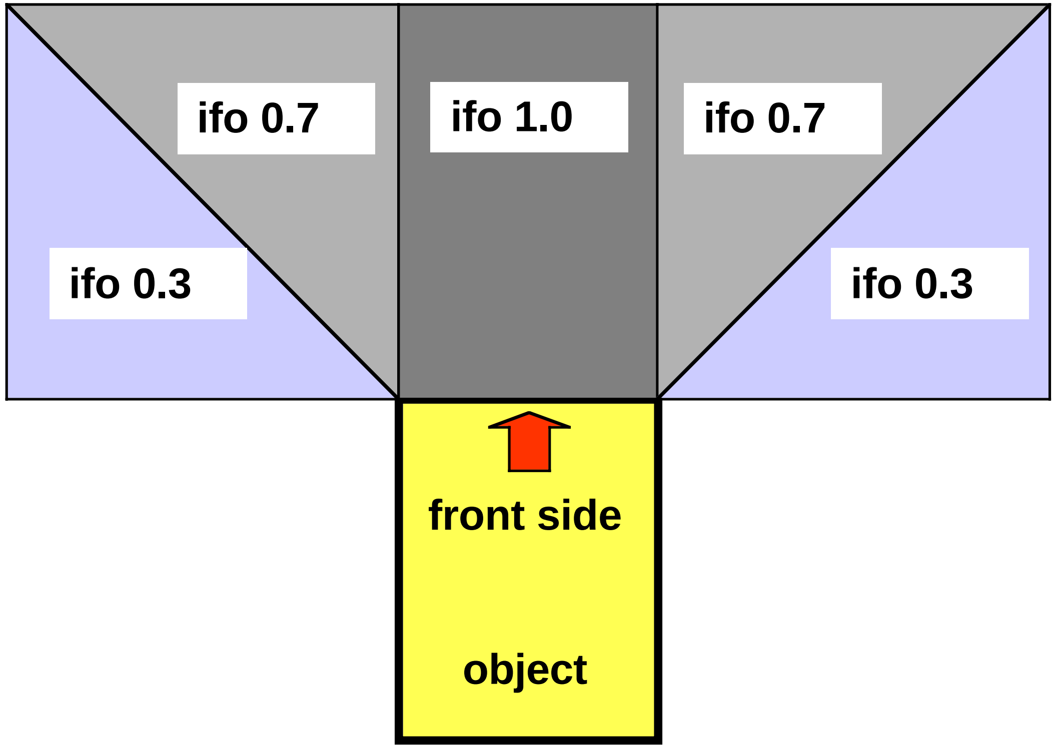
Qualitative relations may ...

- **abstract from quantitative values**	"contains", "touches"
- **express a range of values**	d10: $8 \leq d < 12$
- **express fuzzy relations**	"left-of", "above"
- **enable soft comparisons**	fuzzy-set theory

Qualitative spatial relations

Qualitative spatial relations are expressed by "linguistic variables" (fuzzy variables, symbols with fuzzy values)

<u>Example</u>: "in front of" (ifo)



Combining Fuzzy Propositions

<u>Example</u>: Combining fuzzy spatial relations

"Look for a red light in front of a house and above the entrance"

(light1 in-front-of house1, 0.7) and (light1 above entrance1, 0.4)
(light2 in-front-of house2, 0.5) and (light2 above entrance2, 0.6)

Which light matches the description best?

Formal conjunction of fuzzy values:

$[x, \delta(x)]$, $[y, \delta(y)]$, $0 \leq \delta() \leq 1$ $\delta(x \,\&\, y) = ?$

alternative 1: $\delta(x \,\&\, y) = \delta(x) \cdot \delta(y)$ product of fuzzy values

alternative 2: $\delta(x \,\&\, y) = \min \{\delta(x), \delta(y)\}$ minimum of fuzzy values

Probability theory provides a better foundation for uncertainty management

Recognition of Views by Qualitative 2D-Spatial Relations

Development of "spectacles" for the blind in project MOVIS:

- **spectacles contain 2 mini cameras**
- **blind person may store important views (view models are generated automatically)**
- **view model can be used to recognize a view during walking**

<u>Technical problem</u>:

How can one determine the correspondence of a test view with a model view in spite of

- **changed perspective**
- **changed illumination**
- **changed objects?**

Views of the Same Location from Different Perspectives

Views of the Same Location under Different Illumination

12h

14h

16h

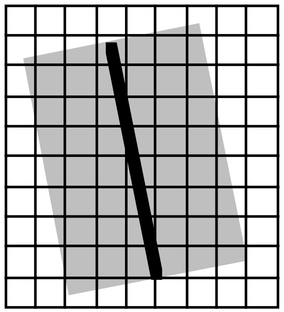
17h

Relational Description of Views

Principle:

- description of views by "interesting" image elements and their spatial relations
- use of straight edges and their properties as "interesting" image elements

straight edge with left and right environment



properties of an edge
(I =intensity,
H = hue
S = saturation):

orientientation:	[..]
length:	[..]
I-mean/variance-left:	(,)
I-mean/variance-right:	(,)
H-mean/variance-left:	(,)
H-mean/variance-right:	(,)
S-mean/variance-left:	(,)
S-mean/variance-right:	(,)
I-contrast:	[-1 .. +1]
H-contrast:	[-1 .. +1]
S-contrast:	[-1 .. +1]
total contrast:	[-1 .. +1]
significance:	[0 .. 1]

Location Relation between Edges

Possible relative locations of 2 edges are described by "offset regions"

For <u>test views</u>:

- **uncertain reference points**

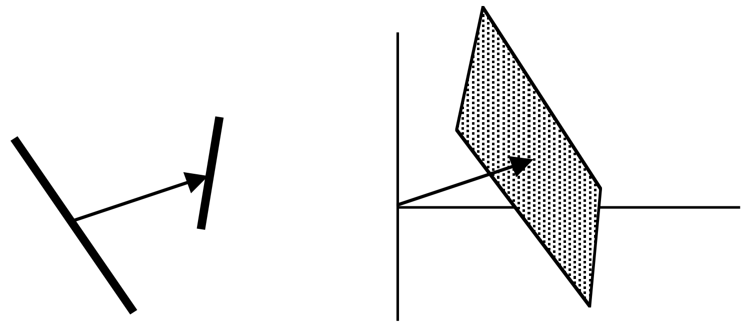


image elements **offset region**

For <u>model views</u>:

- **uncertain reference points**
- **uncertain depth values**
- **uncertain perspective**

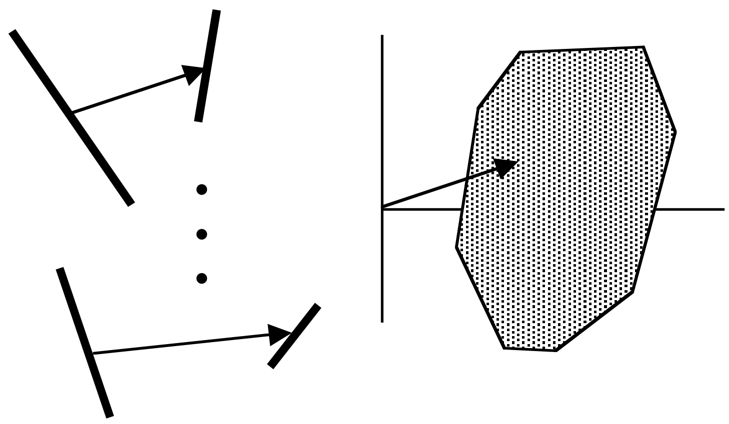
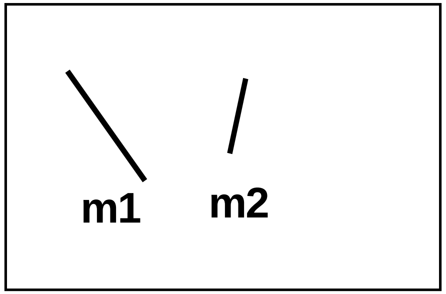


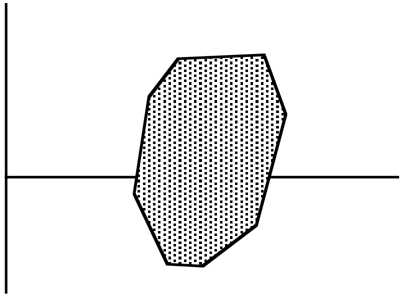
image elements **offset region**

Compatibility Test for Location Relation

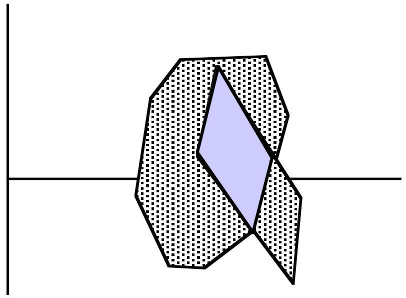
Is the spatial relation of a test pair of edges compatible with the spatial relation of a model pair of edges?



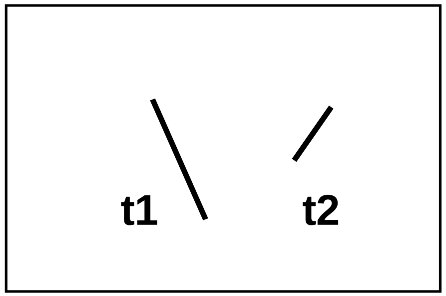
model view



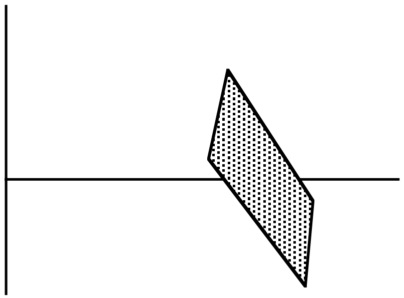
offset region for m1 and m2



compatibility test by intersecting the offset regions (empty = incompatible)

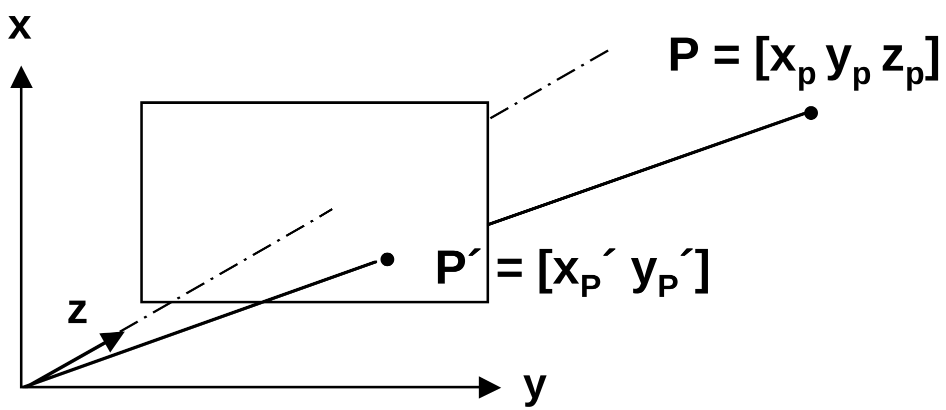


test view



offset region for t1 and t2

Determining Offset Regions



Determine the image of P, if

- **the camera is translated by $\Delta t = [\Delta x, \Delta y, \Delta z]$ and rotated by $[\Delta\alpha, \Delta\beta, \Delta\gamma]$,**
- **the depth of P is given by an uncertainty interval of $[z_{p\text{-}min}\ z_{p\text{-}max}]$**

Perspective projection applied to boundary values of uncertainty intervals provides corner points of <u>offset region</u>.

Offset Regions for Different Uncertainty Intervals

		a	b	c	d
$[\Delta x_{min}\ \Delta x_{max}]$	:	[-1m +1m]	[-1m +1m]	[-1m +1m]	[-1m +1m]
$[z1_{min}\ z1_{max}]$	:	[19m 21m]	[19m 21m]	[9m 51m]	[9m 51m]
$[z2_{min}\ z2_{max}]$	:	[29m 31m]	[9m 51m]	[29m 31m]	[9m 51m]
$[\Delta\gamma_{min}\ \Delta\gamma_{max}]$	:	[-5º +5º]	[-5º +5º]	[-5º +5º]	[-5º +5º]