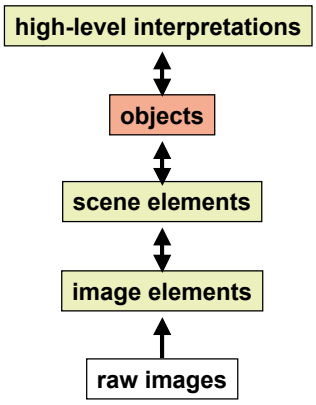
Object Recognition



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 (as proposed in biological vision)

1

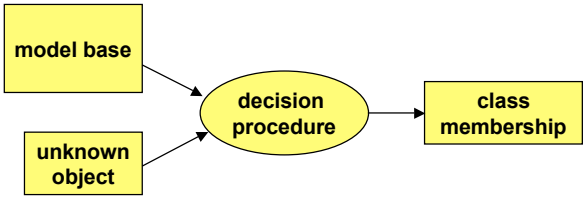
The Chair Room

(H. Bülthoff, MPI Tübingen)

About Model-based Recognition

"model" = generic description of a class of objects

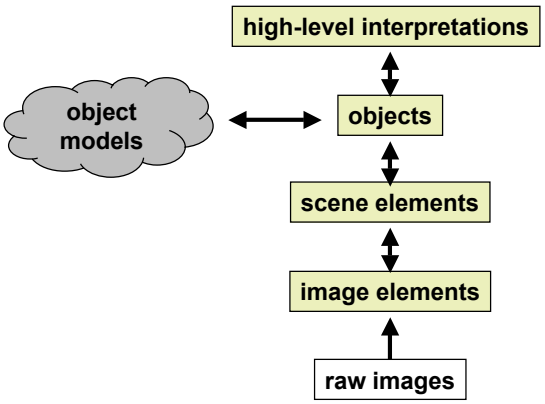
- explicit representation of object properties
  (as opposed to decision procedures which incorporate class properties implicitely)
- generic (class-independent) decision procedure
- reusable and incremental model bases
- no strict correspondence with biological vision



3

Model-based Object Recognition

How to classify objects based on a generic description.

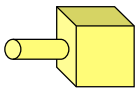


4

3D Models vs. 2D Models

1. Requirement:
Object models must represent invariant class properties
=> 3D models, properties independent of views

e.g.

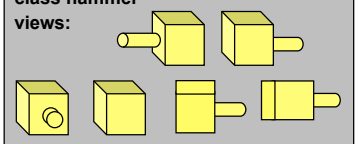


class hammer
is-a aggregat
has-parts part1, part2
is-a part1 cube
is-a part2 cylinder
coaxially-connected part1 part2

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:

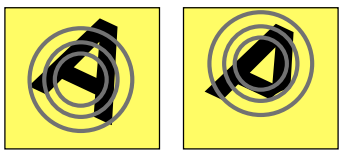


5

Holistic Models vs. Component Models

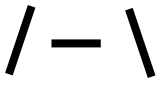
Holistic ("global") models:
- properties refer to complete object
- local disturbances may jeopardize all properties

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:



6

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**

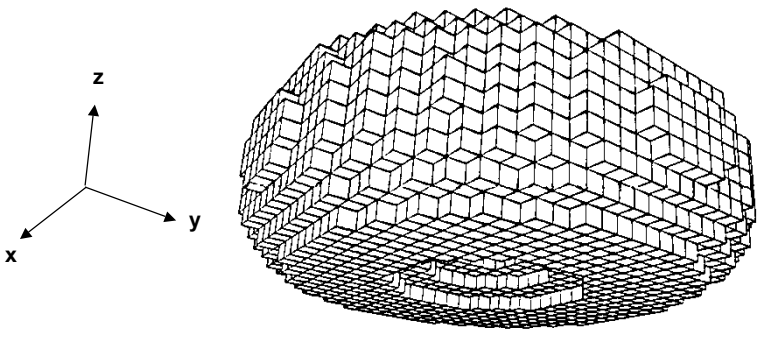
In general, pure 3D models are not immediately useful for Computer Vision because they do not support recognition.

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

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

7

3D Space Occupancy Model



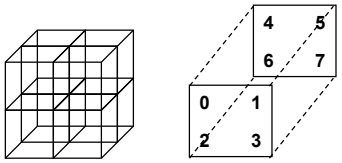
3D shape represented by cube primitives

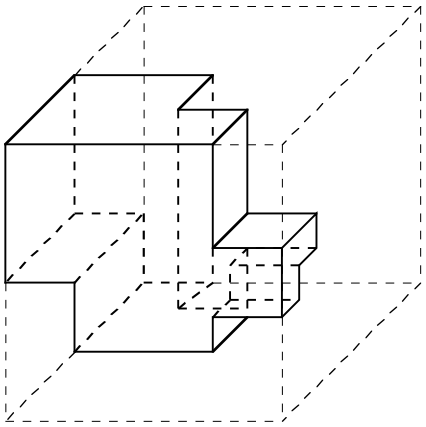
- **useful for highly irregular shapes (e.g. medical domain)**
- **useful for robotics applications (e.g. collision avoidance)**
- **interior cubes do not provide information relevant for views**
- **no explicit surface properties (e.g. surface normals)**

8

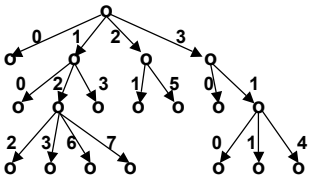
Oct-trees

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





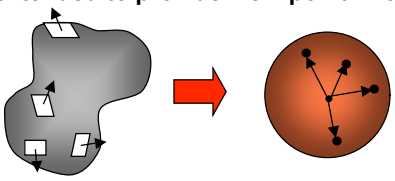
Oct-tree for example (right):



9

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)

10

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:

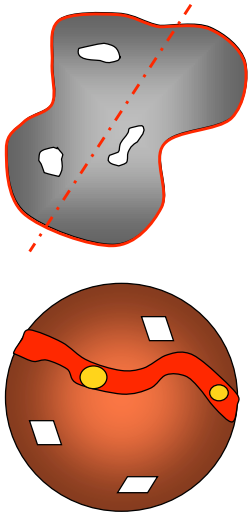
It is assumed that 3D surface normals are determined
(e.g. by laser measurements)

- 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

11

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

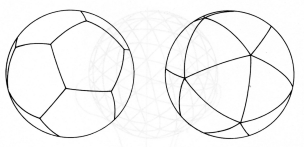


12

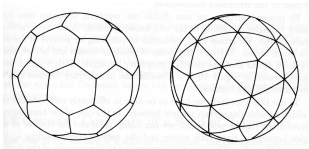
Discretizing the Surface of a Sphere

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.

13

Representing Axial Bodies

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

What representations capture the inherent restrictions of such shapes?

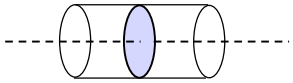
(a)

14

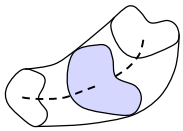
Generalized Cylinders

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

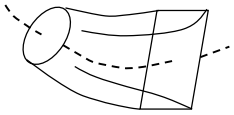
ordinary cylinder swept out by a circle along a straight line



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



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.

15

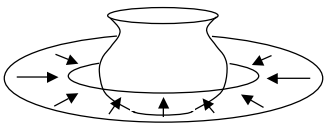
Conditions for 3D Reconstruction from Contours

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)



16

Object Recognition using Relational Matching

17

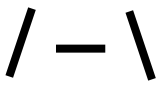
Relational Models

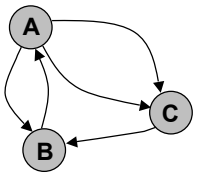
Relational models describe objects (object classes) based on parts (components) and relations between the parts

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

18

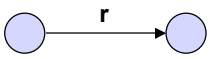
Relations between Components

unary relation: property

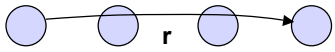
n-ary relation: relation, constraint

Graphical representation

binary relation:



n-ary relation:



"hypergraph"

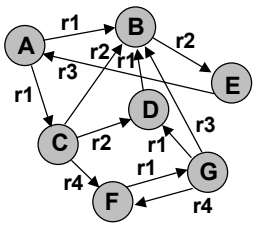
19

Object Recognition by Relational Matching

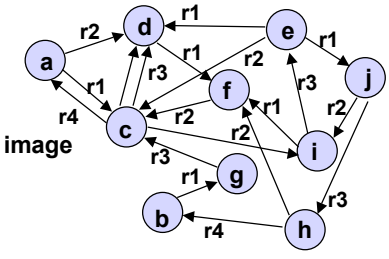
Principle:

- 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



20

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

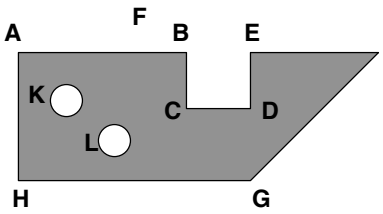
A relational image description B is compatible with a relational model M, if there exists a bijective mapping of nodes of a partial structure B´of B onto nodes of a partial structure M´of M such that

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

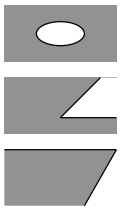
21

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

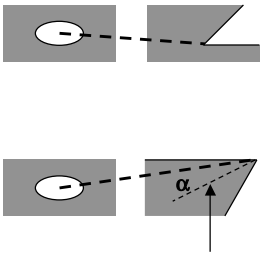
exterior corner

t type T3
w angle

22

Example of a Relational Model (2)

relations between primitive descriptive elements (edges)

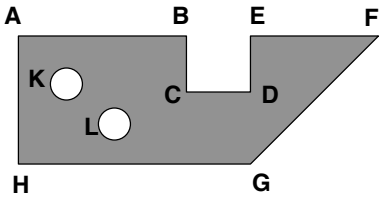


...
d10 distance 10 ± 1
d12 distance 12 ± 1
d14 distance 14 ± 1
...

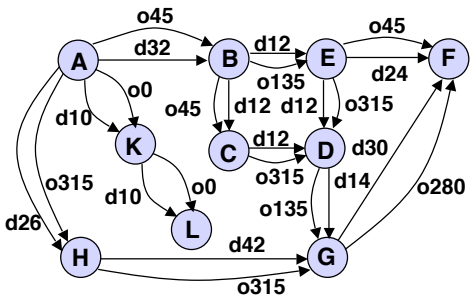
...
o10 orientation 10 ± 5
o20 orientation 20 ± 5
o30 orientation 30 ± 5
...

23

Example of a Relational Model (3)



A t T3 w 90	E t T3 w 90	K t T1 f 48 a 1
B t T3 w 90	F t T3 w 45	K t T1 f 48 a 1
C t T2 w 90	G t T3 w 135	
D t T2 w 90	H t T3 w 90	

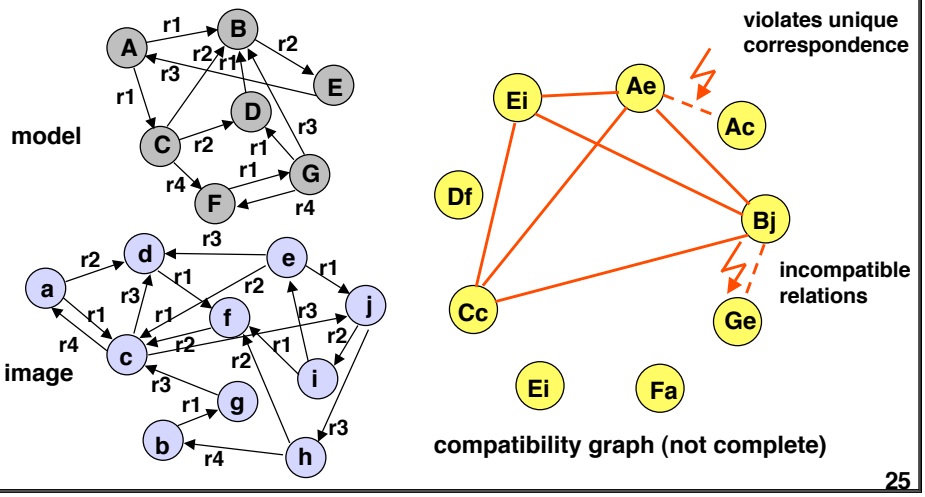


(not all edges are shown)

24

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



25

Finding Maximal Cliques

clique = complete subgraph

Find maximal cliques in a given compatibility graph

Algorithms are available in the literature, e.g.

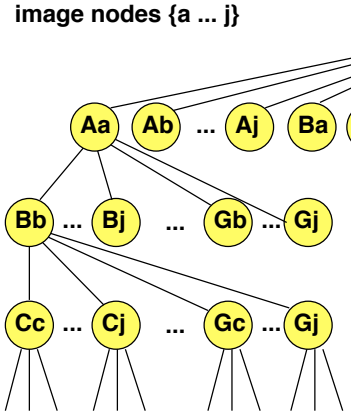
Bron & Kerbusch, Finding all Cliques of an Undirected Graph, Communications of the ACM, Vol. 16, Nr. 9, S. 575 - 577, 1973.

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

26

Relational Matching with Heuristic Search

Stepwise correspondence search between model nodes {A ... G} and image nodes {a ... j}



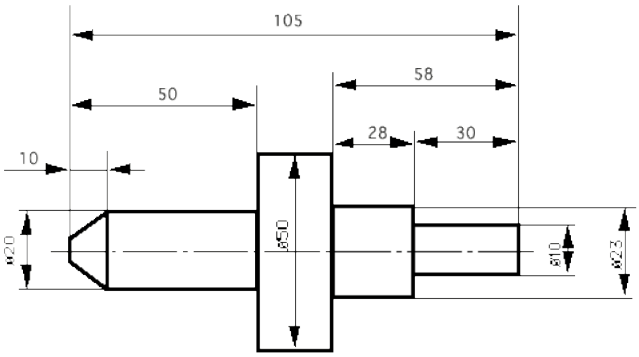
- quality function evaluates partial matches
- accept a partial match if quality > acceptance threshold
- refute a partial match, if quality < refutation threshold

27

Case study: Drawing Interpretation

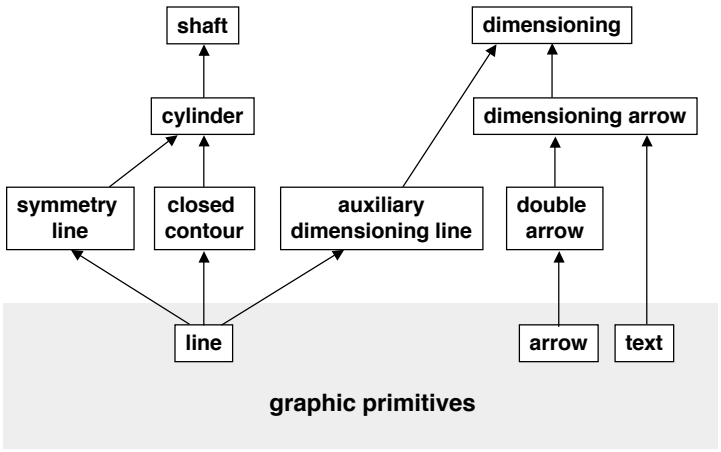
Transforming paper drawings into CAD formats (Pasternak 94)

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



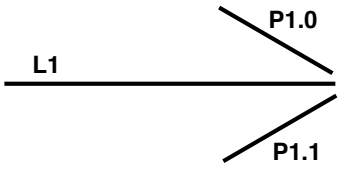
28

Partonomy of Object Parts



29

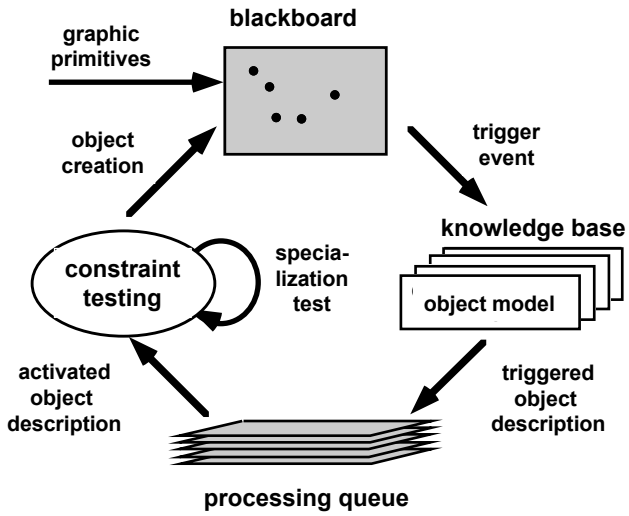
Specification of an Arrow



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

30

Processing Cycle



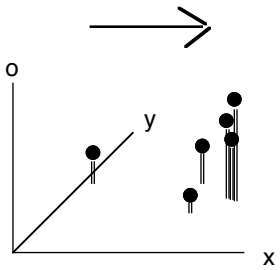
31

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



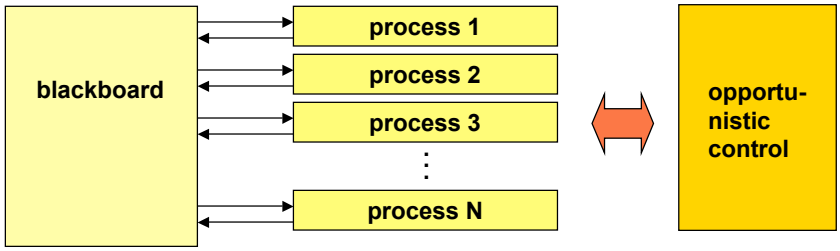
How to construct property spaces:

- **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**

32

Blackboard Architecture

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

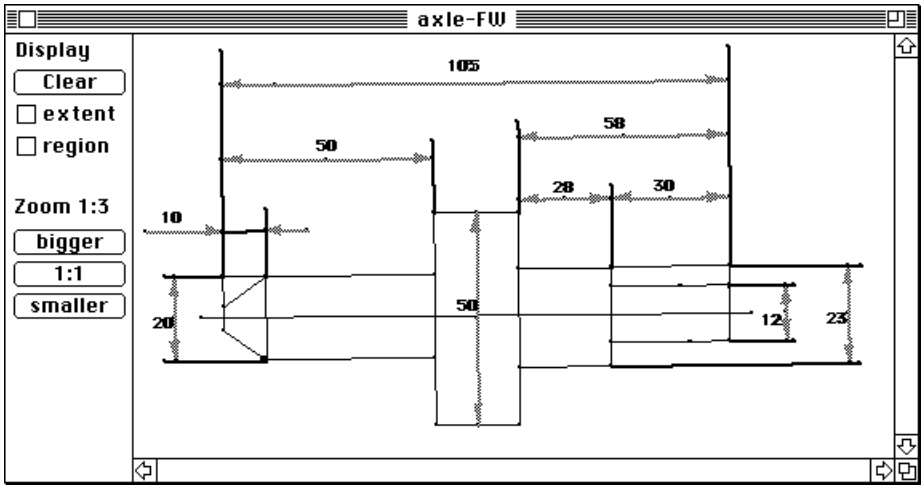


Recognition process may be structured into processes dedicated to the recognition of individual components

33

Analysis of a Machine Drawing

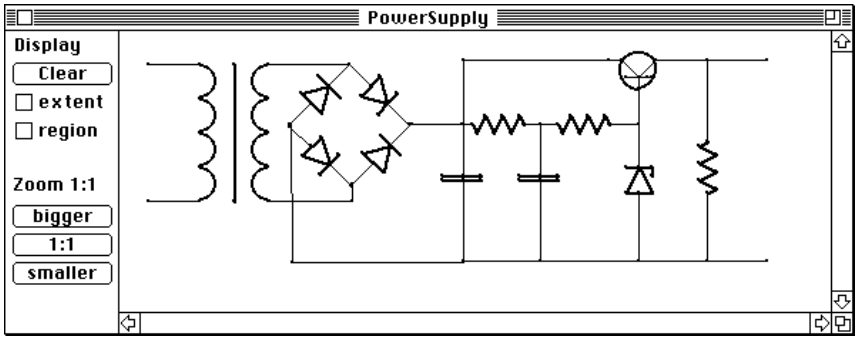
Recognition of dimensioning



34

Analysis of an Electrical Circuit

Recognition of electrical components



35

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 ...

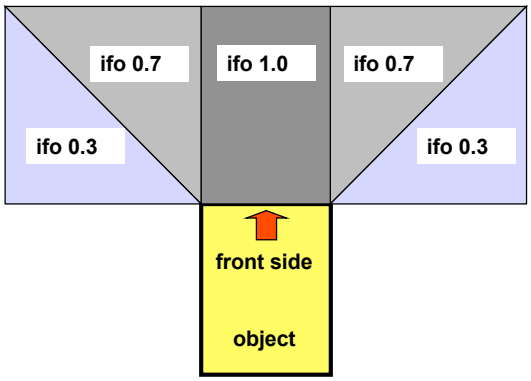
-	**abstract from quantitative values**	"contains", "touches"
-	**express a range of values**	d10: 8 ≤ d < 12
-	**express fuzzy relations**	"left-of", "above"
-	**enable soft comparisons**	fuzzy-set theory

36

Qualitative spatial relations

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

Example: "in front of" (ifo)



37

Combining Fuzzy Propositions

Example: 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:

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

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

38

Qualitative View Recognition

39

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**

Technical problem:

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

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

40

Views of the Same Location from Different Perspectives




41

Views of the Same Location under Different Illumination


12h


14h


16h


17h

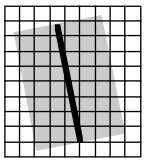
42

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]

43

Location Relation between Edges

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

For test views:

- uncertain reference points

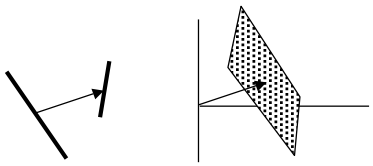


image elements offset region

For model views:

- uncertain reference points
- uncertain depth values
- uncertain perspective

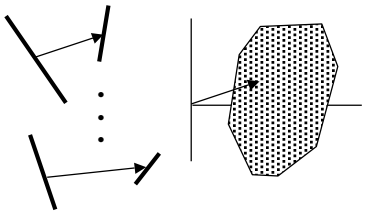
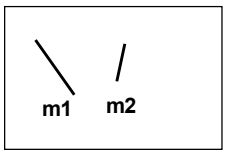


image elements offset region

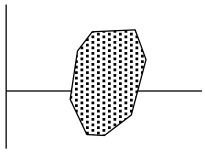
44

Compatibility Test for Location Relation

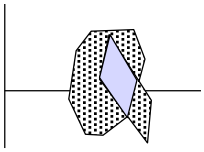
Is the spatial relation of a test pair of edges compatible with the spatial relation of a model pair of edges?



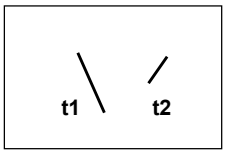
model view



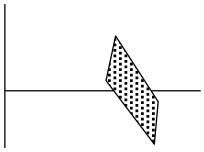
offset region
for m1 and m2



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



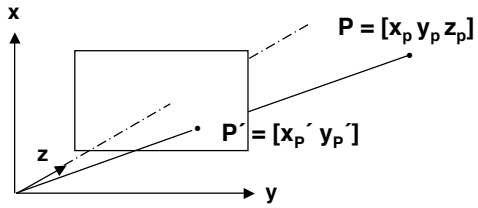
test view



offset region
for t1 and t2

45

Determining Offset Regions



Determine the image of P, if

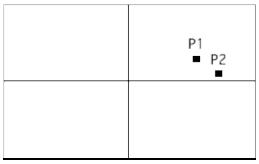
- **the camera is translated by $\Delta t = [\Delta x, \Delta y, \Delta z]$ and rotated by $[\Delta\alpha, \Delta\beta, \Delta\gamma]$,**
- **the depth of P is given by an uncertainty interval of $[z_{p\text{-}min}\ z_{p\text{-}max}]$**

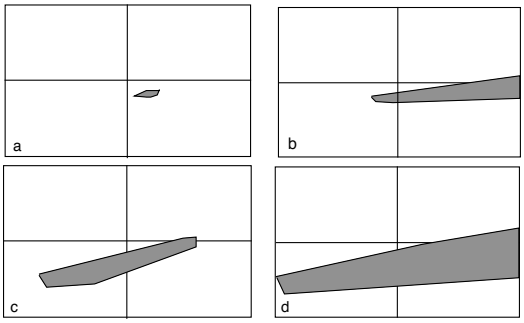
Perspective projection applied to boundary values of uncertainty intervals provides corner points of offset region.

46

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]
$[\Delta\gamma_{min}\ \Delta\gamma_{max}]$	:	[-5° +5°]	[-5° +5°]	[-5° +5°]	[-5° +5°]





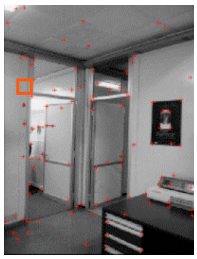
47

Patch-based Object Recognition

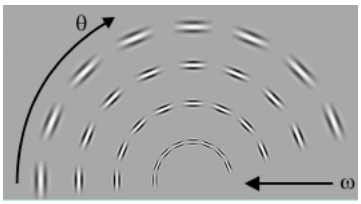
48

Basic Idea

- Determine interest points in image
- Determine local image properties around interest points
- Use local image properties for object classification



Example: Interest points determined by Haralick Operator

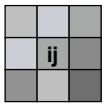


Example: Gabor-Filterbank for local image description

49

Interest Operators (1)

Moravec interest operator: $M(i,j) = \frac{1}{8} \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} |g(m,n) - g(i,j)|$



Zuniga-Haralick operator:

- fit a cubic polynomial

  $f(i,j) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3$

  For a 5x5 neighbourhood the coefficients of the best-fitting polynomial can be directly determined from the 25 greyvalues

- compute interest value from polynomial coefficients

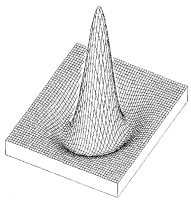
$ZH(i,j) = \frac{-2(c_2^2c_6 - c_2c_3c_5 - c_3^2c_4)}{(c_2^2 + c_3^2)^{\frac{3}{2}}}$ measure of "cornerness" of the polynomial

50

Interest Operators (2)

Difference-of-Gaussians (DoG)

Locates edges at zero crossings of second derivative of smoothed image



"mexican-hat operator"

Harris interest operator:

Determine points with two strong principle curvatures

R = det(H) - k tr(H) = αβ - k (α+β)

α and β are eigenvalues of Hessian matrix H and proportional to main curvatures

$$\mathrm{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

51

SIFT Features

SIFT = Scale Invariant Image Features

David G. Lowe: Distinctive Image Features from Scale-Invariant Keypoints International Journal of Computer Vision, 2004

52

Computation Steps for SIFT Features

1. **Scale-space extrema detection:** The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. **Keypoint localization:** At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

3. **Orientation assignment:** One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. **Keypoint descriptor:** The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

53

Scale Space

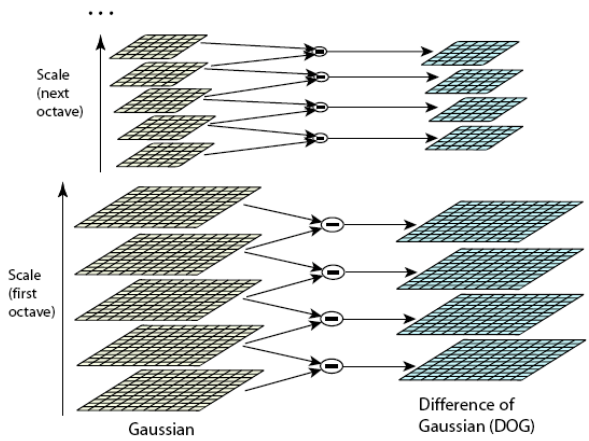
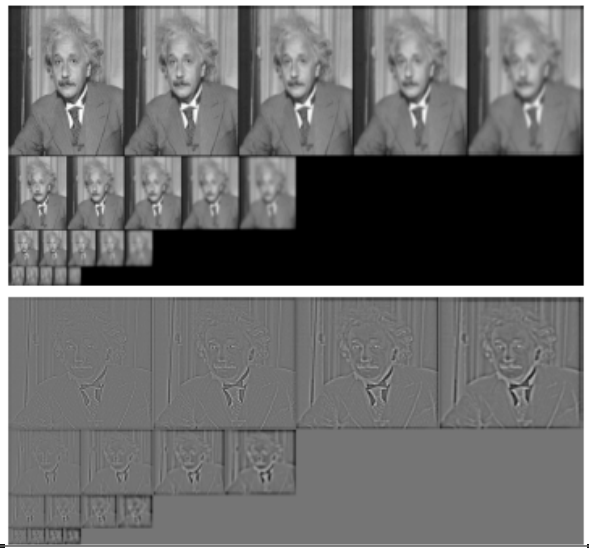


Figure 1: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

54

Example Image in Scale Space



55

Maxima Detection

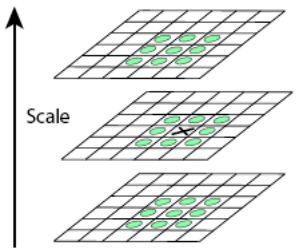


Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

56

Keypoint Selection

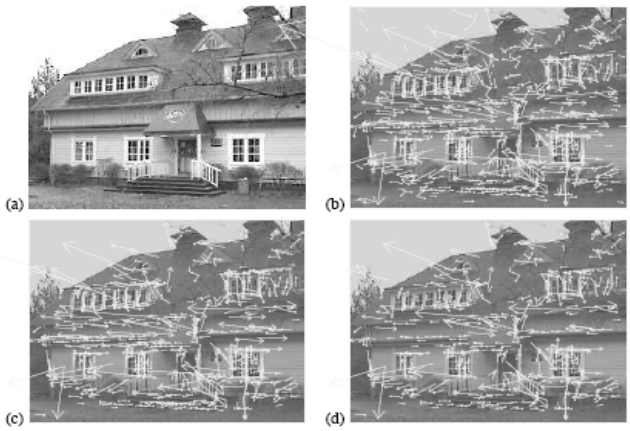


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

57

Eliminating Edge Responses

Compute Hessian at keypoint: $\mathrm{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$

Eigenvalues α and β are proportional to principal curvatures.
Both principle curvatures must be significant for a keypoint to be stable.

Note that

$$\mathrm{Tr}(\mathrm{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$\mathrm{Det}(\mathrm{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

Hence one can check the ratio r = α/β of the principle curvatures by evaluating

$$\frac{\mathrm{Tr}(\mathrm{H})^2}{\mathrm{Det}(\mathrm{H})} < \frac{(r+1)^2}{r}$$

58

Local Image Descriptor

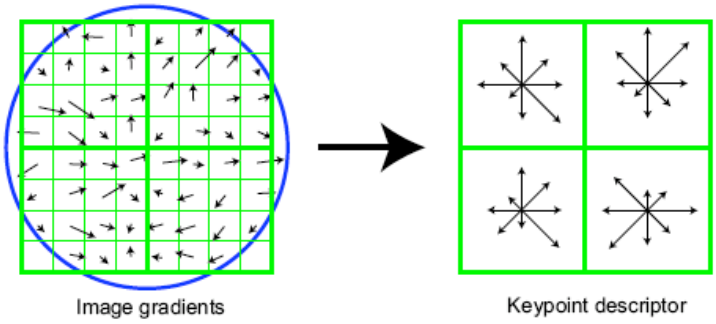


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

59

SIFT Feature Matching

- **Find nearest neighbor in a database of SIFT features from training images.**
- **For robustness, use ratio of nearest neighbor to ratio of second nearest neighbor.**
- **Neighbor with minimum Euclidean distance => expensive search.**
- **Use an approximate, fast method to find nearest neighbor with high probability**

60

Recognition Using SIFT Features

- Compute SIFT features on the input image
- Match these features to the SIFT feature database
- Each keypoint specifies 4 parameters: 2D location, scale, and orientation.
- To increase recognition robustness: Hough transform to identify clusters of matches that vote for the same object pose.
- Each keypoint votes for the set of object poses that are consistent with the keypoint's location, scale, and orientation.
- Locations in the Hough accumulator that accumulate at least 3 votes are selected as candidate object/pose matches.
- A verfication step matches the training image for the hypothesized object/pose to the image using a least-squares fit to the hypothesized location, scale, and orientation of the object.

61

Experiments (1)

Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affi ne transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

62

Experiments (2)

Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affi ne transform used for recognition.

63

SIFT Features Summary

- **SIFT features are reasonably invariant to rotation, scaling, and illumination changes.**
- **They can be used for matching and object recognition (among other things).**
- **Robust to occlusion: as long as we can see at least 3 features from the object we can compute the location and pose.**
- **Efficient on-line matching: recognition can be performed in close-to-real time (at least for small object databases).**

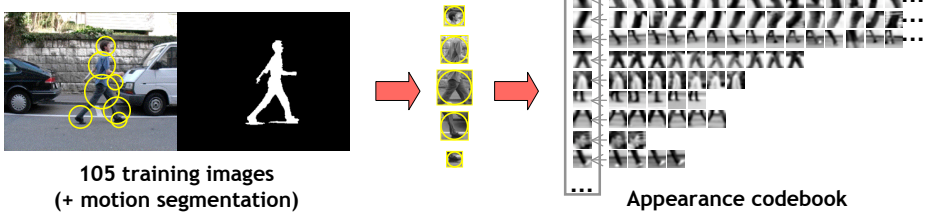
64

Patch-based Object Categorization and Segmentation

Bastian Leibe, Ales Leonardis, and Bernt Schiele:
Combined Object Categorization and Segmentation with an Implicit Shape Model
In ECCV'04 Workshop on Statistical Learning in Computer Vision, Prague, May 2004.

65

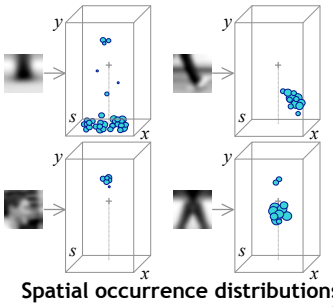
Implicit Shape Model - Representation



105 training images
(+ motion segmentation)

Appearance codebook

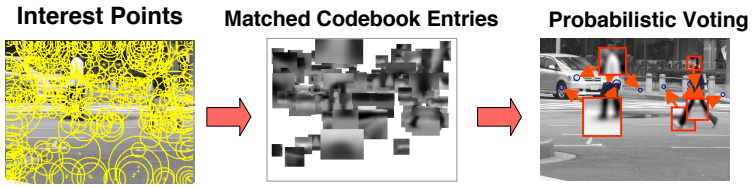
- Learn appearance codebook
  - Extract patches at DoG interest points
  - Agglomerative clustering ⇒ codebook
- Learn spatial distributions
  - Match codebook to training images
  - Record matching positions on object

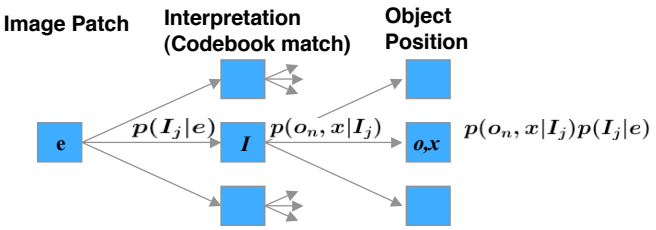


Spatial occurrence distributions

66

Implicit Shape Model - Recognition (1)

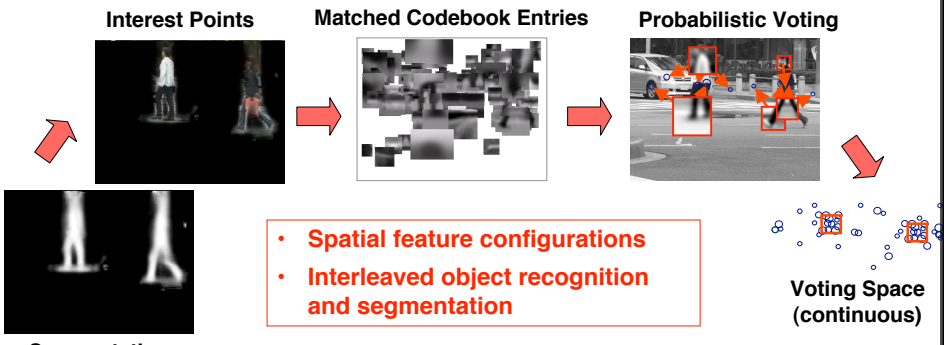




$$p(o_n, x|e) = \sum_j p(o_n, x|I_j)p(I_j|e)$$

67

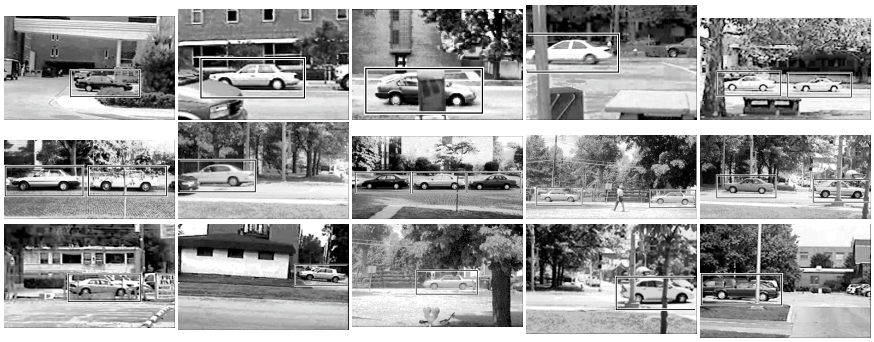
Implicit Shape Model - Recognition (2)



68

Car Detection

- Recognizes different kinds of cars
- Robust to clutter, occlusion, noise, low contrast



69

Cow Detection

- frame-by-frame detection
- no temporal continuity exploited

70