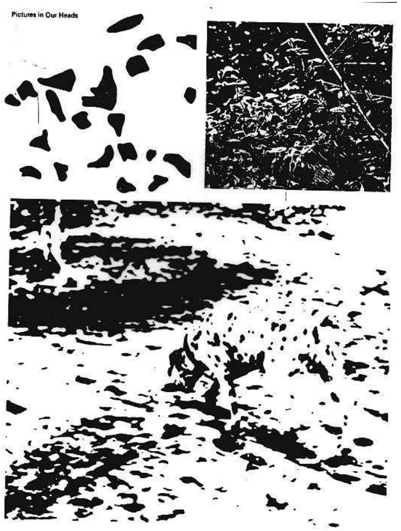
Object Recognition

- REPRESENTATION PROBLEM
  2D & 3D OBJECT MODELING
  SHAPE, COLOR, TEXTURE, ...
- MODEL ACQUISITION PROBLEM
- SEARCH PROBLEM
  HOW TO SELECT IMAGE FEATURES & INDEX INTO DB of ALL OBJECTS TO FIND BEST MATCH ?

⇒ SELECTION/SEGMENTATION < FEATURE DETECTION, PERCEPTUAL GROUPING
INDEXING
MATCHING (SIMILARITY)
CORRESPONDENCE
VERIFICATION



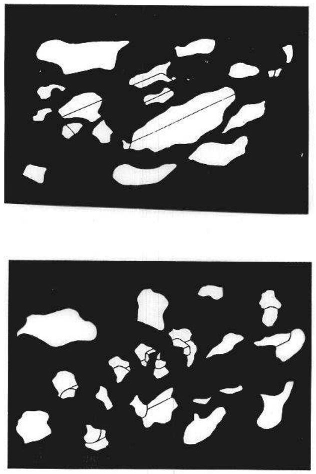
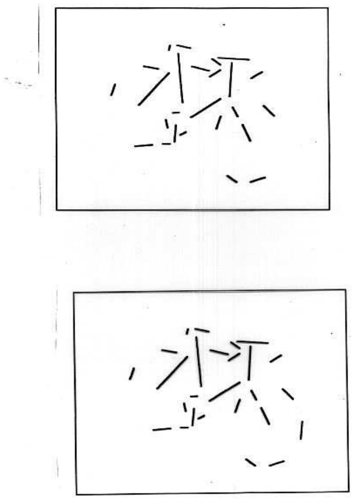
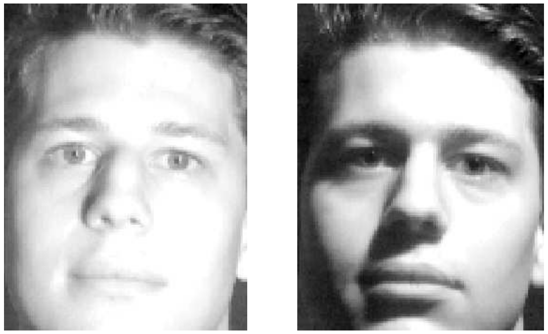
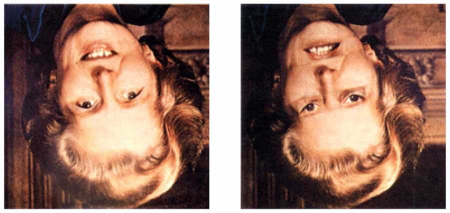


Figure 26. Recoverable version of an object where the contour deletion is produced by an occluding surface (The object, a flashlight, is the same as that shown in Figure 25. The reader may note that the three dimensional percept in this figure does not occur instantaneously.)

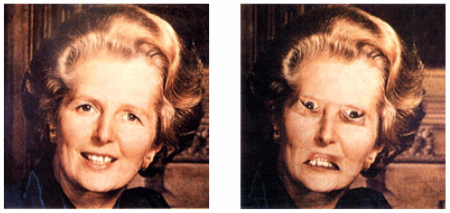




Lighting affects appearance



The “Margaret Thatcher Illusion” by Peter Thompson



The “Margaret Thatcher Illusion” by Peter Thompson

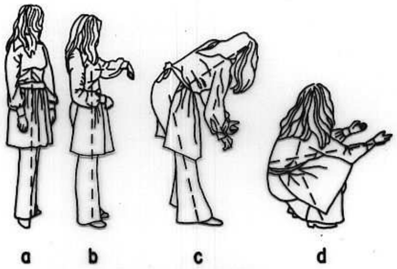
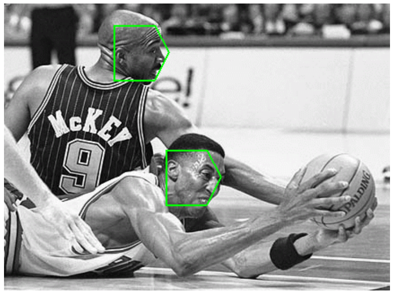


Figure 29. Four configurations of a nonrigid object.

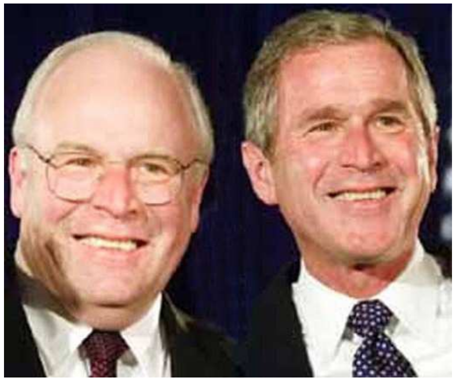
Recognition Problems

- What is it?
  - Object detection
- Who is it?
  - Recognizing identity
    - Object recognition
    - Category recognition
- What are they doing?
  - Activity recognition
- All of these are **classification** problems
  - Choose one class from a list of possible candidates

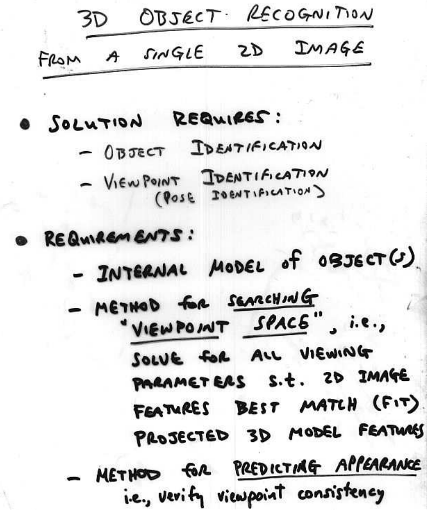
Face Detection



Face Recognition



P. Sinha and T. Poggio, Last but not least, *Perception* **31**, 2002, 133.

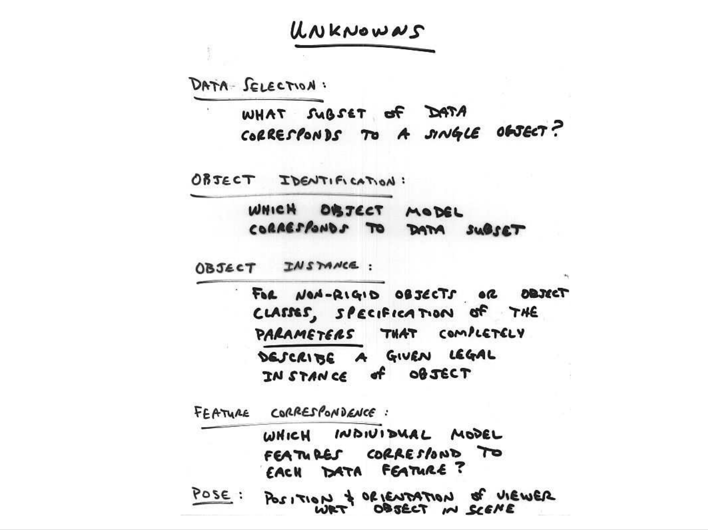


3D OBJECT RECOGNITION FROM A SINGLE 2D IMAGE

- SOLUTION REQUIRES:
  - OBJECT IDENTIFICATION
  - VIEWPOINT IDENTIFICATION (POSE IDENTIFICATION)
- REQUIREMENTS:
  - INTERNAL MODEL of OBJECT(S)
  - METHOD for SEARCHING "VIEWPOINT SPACE", i.e., SOLVE for ALL VIEWING PARAMETERS s.t. 2D IMAGE FEATURES BEST MATCH (FIT) PROJECTED 3D MODEL FEATURES
  - METHOD for PREDICTING APPEARANCE i.e., verify viewpoint consistency

What Makes Recognition Hard?

- Intrinsic variability within each class
- Pose variability
- Illumination variability
- Background variability
- Segmentation problem
  - What region within an image contains the object?
- Feature selection problem
  - What features describe shape and appearance?



UNKNOWNS

DATA SELECTION:

WHAT SUBSET OF DATA CORRESPONDS TO A SINGLE OBJECT?

OBJECT IDENTIFICATION:

WHICH OBJECT MODEL CORRESPONDS TO DATA SUBSET

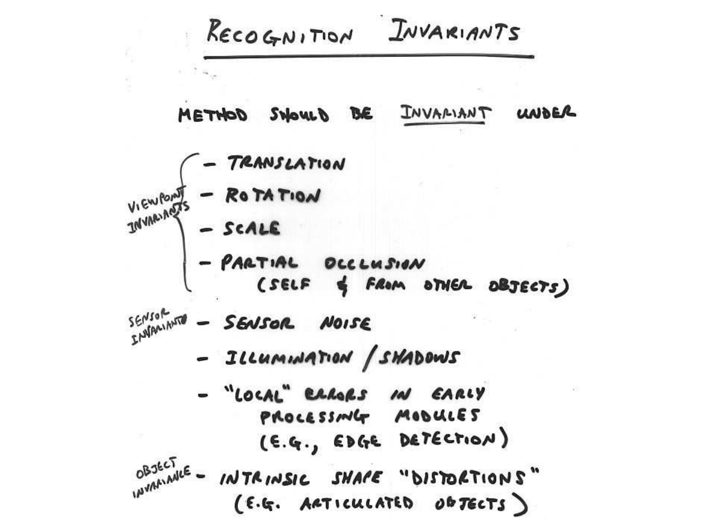
OBJECT INSTANCE:

FOR NON-RIGID OBJECTS OR OBJECT CLASSES, SPECIFICATION OF THE PARAMETERS THAT COMPLETELY DESCRIBE A GIVEN LEGAL INSTANCE OF OBJECT

FEATURE CORRESPONDENCE:

WHICH INDIVIDUAL MODEL FEATURES CORRESPOND TO EACH DATA FEATURE?

POSE: POSITION & ORIENTATION OF VIEWER WRT OBJECT IN SCENE



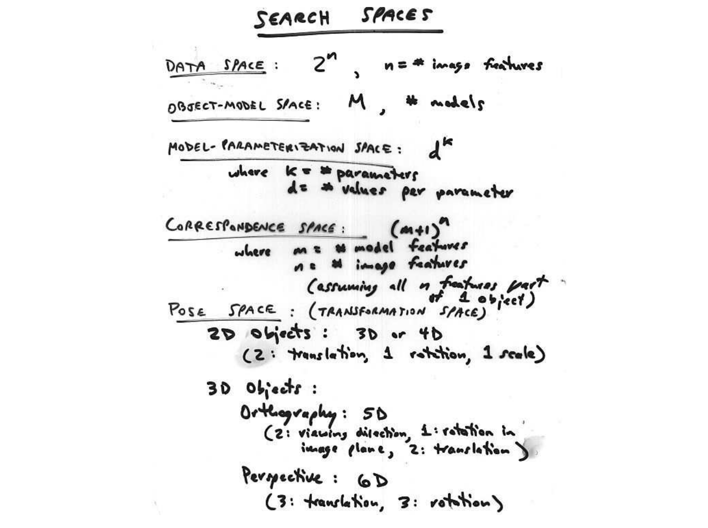
RECOGNITION INVARIANTS

METHOD SHOULD BE INVARIANT UNDER

VIEWPOINT INVARIANTS
- TRANSLATION
- ROTATION
- SCALE
- PARTIAL OCCLUSION (SELF & FROM OTHER OBJECTS)

SENSOR INVARIANTS
- SENSOR NOISE
- ILLUMINATION / SHADOWS
- "LOCAL" ERRORS IN EARLY PROCESSING MODULES (E.G., EDGE DETECTION)

OBJECT INVARIANCE
- INTRINSIC SHAPE "DISTORTIONS" (E.G. ARTICULATED OBJECTS)



SEARCH SPACES

DATA SPACE: 2^n, n = # image features

OBJECT-MODEL SPACE: M, # models

MODEL-PARAMETERIZATION SPACE: d^k

where k = # parameters
d = # values per parameter

CORRESPONDENCE SPACE: $(m+1)^n$

where m = # model features
n = # image features
(assuming all n features part of 1 object)

POSE SPACE: (TRANSFORMATION SPACE)

2D objects: 3D or 4D
(2: translation, 1 rotation, 1 scale)

3D Objects:

Orthography: 5D
(2: viewing direction, 1: rotation in image plane, 2: translation)

Perspective: 6D
(3: translation, 3: rotation)

MATCHING MODELS w/ IMAGES

2D-2D MATCH

- SMALL # "CHARACTERISTIC VIEWS"
  - MAY BE USEFUL for FAMILIAR OBJECTS w/ FEW "TYPICAL" VIEWS
- VIEWER-CENTERED MODELS

2½D/3D-3D MATCH

- RECOVER **BOTTOM-UP** FROM IMAGE(S) 2½D or 3D DATA

  E.G. STEREO
  SHAPE-FROM X
  GEONS [Biederman]
  GENERALIZED CYLINDERS [Marr]

2D-3D MATCH

- OBJECT-CENTERED MODELS
- VIEWPOINT-INVARIANT FEATURES
  OR VIEWPOINT-VARIANT FEATURES

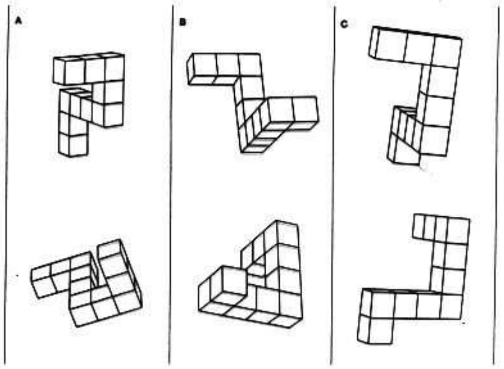
METHOD 0: DIRECT APPROACH

- FOR EACH ~~OBJECT~~ MODEL, STORE A SUFFICIENTLY LARGE NUMBER of DIFFERENT VIEWS
- DEFINE A **SIMILARITY MEASURE** TO FIND BEST MATCH BETWEEN IMAGE DATA AND DIFFERENT VIEWS of ALL OBJECT MODELS

  E.G. CROSS-CORRELATION w/ A TEMPLATE
- DISADVANTAGES:
  - VERY LARGE SPACE of POSSIBLE VIEWS (TRANSFORMATIONS)
  - BRUTE-FORCE SEARCH of "POSE SPACE"
  - OBJECTS ~~CAN~~ SHOULD BE RECOGNIZED IN NOVEL VIEWS

Ex. 2D position & 2D orientation invariant to 1 pixel and 1° ⇒

$512 \times 512 \times 360 \approx 90{,}000{,}000$ poses

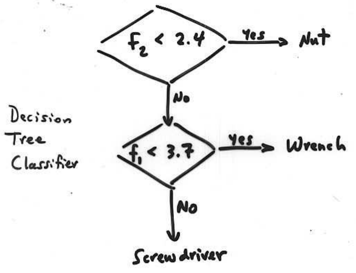


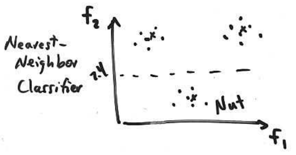
METHOD 1: INVARIANT GLOBAL PROPERTIES

- SELECT A SET of GLOBAL, INVARIANT PROPERTIES — EXPLOITS REGULARITIES ACROSS VIEWPOINTS — VIEWPOINT-INVARIANT FEATURES

  E.G. COMPACTNESS (P^2/A)
  AREA (A), MOMENTS ($\sum\sum x^p y^q f(x,y)$)
- FEATURES MUST BE RELATIVELY **SIMPLE** TO COMPUTE ⇒ NEED PROCEDURE for DETECTING/COMPUTING EACH SUCH PROPERTY
- IF OBJECT'S PROPERTIES HAVE CHARACTERISTIC **RANGES** of VALUES ⇒ POLYTOPE IN n-D FEATURE SPACE
- PATTERN RECOGNITION
  GIBSON
  SRI VISION MODULE
- ARE THERE INVARIANT FEATURES for REAL, 3D OBJECTS ??

Given n-D feature vector, $<f_1, f_2, \ldots, f_n>$, match w/ N possible objects :





A Geometric Invariant

CROSS RATIO

Given 4 points (x_i, y_i, z_i) that are collinear in scene

let $(x_{i_k}, y_{i_k}, z_{i_k})$ be the perspective projection of those 4 points into an image.

$$\text{Cross Ratio} = \frac{(x_3 - x_1)(x_4 - x_2)}{(x_3 - x_2)(x_4 - x_1)}$$

$$= \frac{(x_{i_3} - x_{i_1})(x_{i_4} - x_{i_2})}{(x_{i_3} - x_{i_2})(x_{i_4} - x_{i_1})}$$

and similarly for y's.

⇒ If we know coords of 3 collinear points, we can compute coords of any other point on same line from its image coords

Approximate Invariants

- Over a limited range of viewpoint variation
- Parallelism
- Collinearity
- Angle between a pair of lines
- Co-termination

MATCHING USING OCCLUDING CONTOURS

Kriegman & Ponce, 1990, PAMI

- Given: Smooth, 3D Shape
- Calculate (offline) the implicit equation of the occluding contour
  $$F(u, v, V) = 0$$
- (Foreach aspect)
  Solve for pose by minimizing average distance between model contour and image edges
  $$\min \sum_i F^2(u_i, v_i, V)$$
  where (u_i, v_i) are image edge pts.

(Nonlinear least-squares minimization of an expression for the distance b/w contour & image ⇒ Iterative techniques)

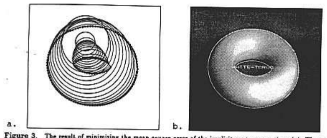


Figure 3. The result of minimizing the mean square error of the implicit contour equation: (a). The contours at each iteration of the minimization are shown overlaid on the edge points used in the minimization; the Canny edgels are drawn as little circles. (b). The result of recognition for the white torus.

To conclude, let us give an example. Consider a torus. The expression C is too complex to be given here. With the following substitutions: $\bar{x} = \bar{x}\sin\beta$, $\bar{y} = \bar{y}\sin\beta$, $c = \cos^2\beta$, the expression C_1 is:

$$
\begin{aligned}
&0 = C_1(\bar{x}, \bar{y}, c) = \\
&R^8c^8 - 6R^8c^7 + 15R^8c^6 - 20R^8c^5 + 15R^8c^4 - 6R^8c^3 + R^8c^2 - 2R^6c^8r^2 + 10R^6c^7r^2 \\
&+2R^6c^7\bar{x}^2 - 18R^6c^6r^2 - 12R^6c^6\bar{x}^2 + 4R^6c^6\bar{y}^2 + 10R^6c^5r^2 + 28R^6c^5\bar{x}^2 - 18R^6c^5\bar{y}^2 \\
&+10R^6c^4r^2 - 32R^6c^4\bar{x}^2 + 32R^6c^4\bar{y}^2 - 18R^6c^3r^2 + 18R^6c^3\bar{x}^2 - 28R^6c^3\bar{y}^2 + 10R^6c^2r^2 \\
&-4R^6c^2\bar{x}^2 + 12R^6c^2\bar{y}^2 - 2R^6cr^2 - 2R^6c\bar{y}^2 + R^4c^8r^4 - 2R^4c^7r^4 + 2R^4c^7r^2\bar{x}^2 - 8R^4c^6r^4 \\
&-10R^4c^6r^2\bar{x}^2 - 6R^4c^6r^2\bar{y}^2 + R^4c^6\bar{x}^4 + 34R^4c^5r^4 + 26R^4c^5r^2\bar{x}^2 + 26R^4c^5r^2\bar{y}^2 - 8R^4c^5\bar{x}^4 \\
&+6R^4c^5\bar{x}^2\bar{y}^2 - 50R^4c^4r^4 - 44R^4c^4r^2\bar{x}^2 - 46R^4c^4r^2\bar{y}^2 + 19R^4c^4\bar{x}^4 - 22R^4c^4\bar{x}^2\bar{y}^2 \\
&+6R^4c^4\bar{y}^4 + 34R^4c^3r^4 + 46R^4c^3r^2\bar{x}^2 + 44R^4c^3r^2\bar{y}^2 - 18R^4c^3\bar{x}^4 + 32R^4c^3\bar{x}^2\bar{y}^2 - 18R^4c^3\bar{y}^4 \\
&-8R^4c^2r^4 - 26R^4c^2r^2\bar{x}^2 - 26R^4c^2r^2\bar{y}^2 + 6R^4c^2\bar{x}^4 - 22R^4c^2\bar{x}^2\bar{y}^2 + 19R^4c^2\bar{y}^4 - 2R^4cr^4 \\
&+6R^4cr^2\bar{x}^2 + 10R^4cr^2\bar{y}^2 + 6R^4c\bar{x}^2\bar{y}^2 - 8R^4c\bar{y}^4 + R^4r^4 - 2R^4r^2\bar{y}^2 + R^4\bar{y}^4 - 2R^2c^7r^6 \\
&+10R^2c^6r^6 - 6R^2c^6r^4\bar{x}^2 + 2R^2c^6r^4\bar{y}^2 - 18R^2c^5r^6 + 26R^2c^5r^4\bar{x}^2 - 10R^2c^5r^4\bar{y}^2 \\
&-6R^2c^5r^2\bar{x}^4 + 4R^2c^5r^2\bar{x}^2\bar{y}^2 + 10R^2c^4r^6 - 46R^2c^4r^4\bar{x}^2 + 26R^2c^4r^4\bar{y}^2 + 22R^2c^4r^2\bar{x}^4 \\
&-14R^2c^4r^2\bar{x}^2\bar{y}^2 - 6R^2c^4r^2\bar{y}^4 - 2R^2c^4\bar{x}^6 + 2R^2c^4\bar{x}^4\bar{y}^2 + 10R^2c^3r^6 + 44R^2c^3r^4\bar{x}^2 \\
&-44R^2c^3r^4\bar{y}^2 - 32R^2c^3r^2\bar{x}^4 + 10R^2c^3r^2\bar{x}^2\bar{y}^2 + 22R^2c^3r^2\bar{y}^4 + 6R^2c^3\bar{x}^6 - 4R^2c^3\bar{x}^4\bar{y}^2 \\
&+6R^2c^3\bar{x}^2\bar{y}^4 - 18R^2c^2r^6 - 26R^2c^2r^4\bar{x}^2 + 46R^2c^2r^4\bar{y}^2 + 22R^2c^2r^2\bar{x}^4 + 10R^2c^2r^2\bar{x}^2\bar{y}^2 \\
&-32R^2c^2r^2\bar{y}^4 - 4R^2c^2\bar{x}^6 + 8R^2c^2\bar{x}^4\bar{y}^2 - 8R^2c^2\bar{x}^2\bar{y}^4 + 4R^2c^2\bar{y}^6 + 10R^2cr^6 + 10R^2cr^4\bar{x}^2 \\
&-26R^2cr^4\bar{y}^2 - 6R^2cr^2\bar{x}^4 - 14R^2cr^2\bar{x}^2\bar{y}^2 + 22R^2cr^2\bar{y}^4 - 6R^2c\bar{x}^4\bar{y}^2 + 4R^2c\bar{x}^2\bar{y}^4 - 6R^2c\bar{y}^6 \\
&-2R^2r^6 - 2R^2r^4\bar{x}^2 + 6R^2r^4\bar{y}^2 + 4R^2r^2\bar{x}^2\bar{y}^2 - 6R^2r^2\bar{y}^4 - 2R^2\bar{x}^2\bar{y}^4 + 2R^2\bar{y}^6 + c^6r^8 - 6c^5r^8 \\
&+4c^5r^6\bar{x}^2 + 2c^5r^6\bar{y}^2 + 15c^4r^8 - 18c^4r^6\bar{x}^2 - 12c^4r^6\bar{y}^2 + 6c^4r^4\bar{x}^4 + 6c^4r^4\bar{x}^2\bar{y}^2 + c^4r^4\bar{y}^4 \\
&-20c^3r^8 + 32c^3r^6\bar{x}^2 + 28c^3r^6\bar{y}^2 - 18c^3r^4\bar{x}^4 - 22c^3r^4\bar{x}^2\bar{y}^2 - 8c^3r^4\bar{y}^4 + 4c^3r^2\bar{x}^6 \\
&+6c^3r^2\bar{x}^4\bar{y}^2 + 2c^3r^2\bar{x}^2\bar{y}^4 + 15c^2r^8 - 28c^2r^6\bar{x}^2 - 32c^2r^6\bar{y}^2 + 19c^2r^4\bar{x}^4 + 32c^2r^4\bar{x}^2\bar{y}^2 \\
&+19c^2r^4\bar{y}^4 - 6c^2r^2\bar{x}^6 - 8c^2r^2\bar{x}^4\bar{y}^2 - 4c^2r^2\bar{x}^2\bar{y}^4 - 2c^2r^2\bar{y}^6 + c^2\bar{x}^8 + 2c^2\bar{x}^6\bar{y}^2 \\
&+c^2\bar{x}^4\bar{y}^4 - 6cr^8 + 12cr^6\bar{x}^2 + 18cr^6\bar{y}^2 - 8cr^4\bar{x}^4 - 22cr^4\bar{x}^2\bar{y}^2 - 18cr^4\bar{y}^4 + 2cr^2\bar{x}^6 \\
&+4cr^2\bar{x}^4\bar{y}^2 + 8cr^2\bar{x}^2\bar{y}^4 + 6cr^2\bar{y}^6 + 2c\bar{x}^6\bar{y}^2 + 4c\bar{x}^4\bar{y}^4 + 2c\bar{x}^2\bar{y}^6 + r^8 - 2r^6\bar{x}^2 - 4r^6\bar{y}^2 \\
&+r^4\bar{x}^4 + 6r^4\bar{x}^2\bar{y}^2 + 6r^4\bar{y}^4 - 2r^2\bar{x}^4\bar{y}^2 - 6r^2\bar{x}^2\bar{y}^4 - 4r^2\bar{y}^6 + \bar{x}^4\bar{y}^4 + 2\bar{x}^2\bar{y}^6 + \bar{y}^8
\end{aligned}
$$

METHOD 2: OBJECT DECOMPOSITION INTO PARTS

(+ CORRESPONDENCE SPACE SEARCH)

- DECOMPOSE OBJECT INTO CONSTITUENT **PARTS**, WHERE EACH PART IS SIMPLE, GENERIC

  E.G., 1D CONTOURS — LINES, CORNERS, "CODONS"

  2D SURFACE PATCHES — HOLES

  3D VOLUMES — GENERALIZED CYLINDERS, "GEONS"

- DECOMPOSITION DONE INDEPENDENT of VIEWPOINT

- 1. SEGMENT IMAGE INTO PARTS
  2. CLASSIFY PARTS
  3. DESCRIBE OBJECT IN TERMS of PARTS
     - SET of FEATURES
     - SPATIAL RELATIONS BETWEEN PARTS

- EXAMPLES:

  PERCEPTRON

  BOLLES' LOCAL-FEATURE FOCUS METHOD

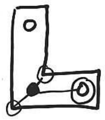
Pose consistency

- Correspondences between image features and model features are not independent
- A small number of correspondences yields a camera --- the others must be consistent with this
- Strategy:
  – Generate hypotheses using small numbers of correspondences (e.g. triples of points for a calibrated perspective camera, etc., etc.)
  – Backproject and verify
- Notice that the main issue here is camera calibration
- Appropriate groups are "frame groups"

BOLLES' LOCAL FEATURE FOCUS METHOD

- 2D LOCAL FEATURES: HOLES, CONVEX CORNERS, CONCAVE CORNERS
- DESCRIBE EACH OBJECT AS A GRAPH
- FOREACH OCCURRENCE of EACH FEATURE TYPE IN EACH MODEL,

  SELECT SUBGRAPH "CENTERED" AT NODE WHICH IS SUFFICIENT TO DISTINGUISH IT FROM ALL OTHERS
- COMPILE A TABLE FOREACH FEATURE TYPE WHICH SAYS WHERE TO LOOK FOR OTHER NEARBY FEATURES IN ORDER TO CONFIRM IT.



Ex. FOCUS - FEATURE TABLE

"FOCUS" FEATURE: HOLE (i.e., CIRCLE)

POSSIBLE OBJECT FEATURES: HOLE1 in OBJECT3
HOLE4 in OBJECT5
HOLE6 in OBJECT 5
⋮

NEARBY FEATURES:

CORNER	DISTANCE, ORIENTATION	CORNER 3 IN OBJ 2
CORNER	DISTANCE, ORIENTATION	CORNER 1 in OBJ 5
HOLE	DISTANCE	HOLE 1 in OBJ 2 OR HOLE 5 in OBJ 4 OR ...

- BUILD A GRAPH FROM NODE-NODE MATCHES —

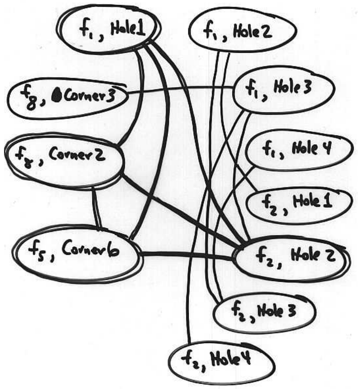
  NODE = (MODEL NODE, IMAGE FEATURE) PAIR

  ARC = CONNECTS PAIRS OF NODES WHICH ARE NEIGHBORS IN MODEL AND IMAGE FEATURES ARE DISTINCT
  (⇒ LOCALLY CONSISTENT (PAIRWISE) ASSIGNMENT)

- FIND MAXIMAL CLIQUE — LARGEST CONNECTED SUBGRAPH
  = LARGEST CLUSTER OF MUTUALLY-CONSISTENT MATCHES

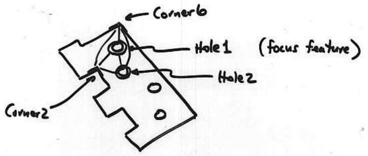
- VERIFY MATCH BY COMPARING COMPLETE MODEL W/ IMAGE

Graph of pairwise-consistent feature matches



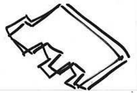
- MAXIMAL CLIQUE IN RED.
- EXAMPLE OF SEARCHING THE "CORRESPONDENCE SPACE"

Hypothesized Solution:



Verification:

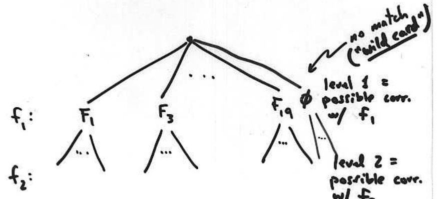
1. Compute pose
2. Project model to image space
3. Cross-correlation



Interpretation Trees

Using geometric constraints to limit search of Correspondence Space

Given: Image features $f_1, \ldots, f_n$
Model features $F_1, \ldots, F_m$


no match ("wild card")
f_1:
f_2:
F_1
F_3
F_{19}
ϕ
level 1 = possible corr. w/ f_1
level 2 = possible corr. w/ f_2 and consistent w/ ancestor correspondences

Depth-first search w/ Pruning at nodes specifying an inconsistent interpretation.

Size of Correspondence Space =

$(m+1)^n$

Height of Interpretation Tree = n

Goal: find path from root to leaf such that all n correspondences are consistent w/ a single model

Approach: Use 1st and 2nd order geometric constraints to prune interpretation tree as we traverse it.

Possible Constraints: Unary Constraints: properties of f_i and F_j must be similar (i.e. match)

Ex. length of edge
corner angle
color
texture

Binary Constraints: consistency constraints between (f_{i_1}, F_{j_1}) and (f_{i_2}, F_{j_2}) pairings.

Ex: Relative angle between 2 edges

Range of distances between 2 edges

Range of directions from f_{i_1} to f_{i_2}

etc.

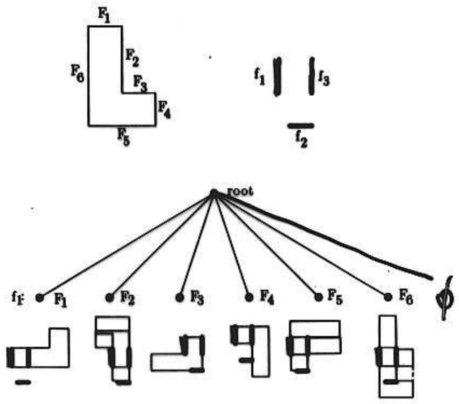


Figure 3.1
We can build a tree of possible interpretations, by first considering all the ways of matching the first data feature, f_1, to each of the model features, $F_j, j = 1, \ldots, m$. In the bottom part of the figure, we show an example of these pairings for the model and data shown at the top. In some cases, due to occlusion of the data features, a range of possible poses is given.

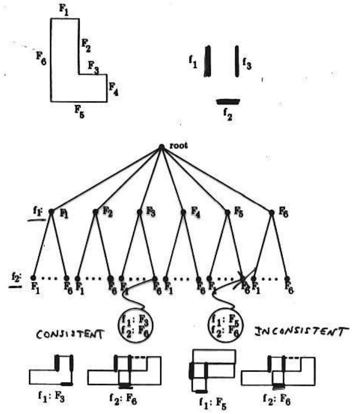


Figure 3.2
For each pairing of the first data feature with a model feature, we can consider matchings for the second data feature with each of the model features. Thus, each node in the second level of the tree defines a pairing for the first two data features, found by tracing up the tree to the root. Two examples are shown. The example on the left is consistent with a single rigid transformation, as shown by the fact that the two ranges of poses specified by each of the data-model feature pairings have a common pose. The example on the right is not consistent with a single transformation, as its two ranges of poses do not have a common pose.

Interpretation Tree Search Algorithm

- Depth-First search with pruning (cut-offs)
- At each node, test all unary & binary constraints. If all are satisfied, continue. Otherwise, backtrack
- At leaf node have an hypothesis for a feasible interpretation. Verify by solving for pose given correspondences, and compare projected model w/ data.

METHOD 3: COMBINING LOCAL EVIDENCE + POSE SPACE SEARCH

- FIRST MATCH, THEN FIND BEST VIEWING TRANSFORMATION
- DEFINE 6D PARAMETER SPACE - POSE SPACE of ALL POSSIBLE (DISCRETELY SAMPLED) VIEWING TRANSFORMATIONS
- 1. FOR EACH MATCHING (IMAGE FEATURE, MODEL FEAT.) PAIR, "VOTE" IN PARAMETER SPACE FOR ALL TRANSFORMATIONS THEY DESCRIBE.

  2. FIND "PEAK" IN PARAMETER SPACE.
- EXAMPLES :

  GENERALIZED HOUGH TRANSFORM

  THOMPSON & MUNDY, 1987

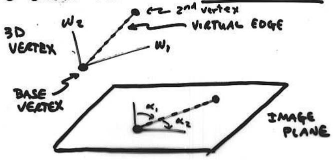
  HINTON, 1981

Voting on Pose

- Each model leads to many correct sets of correspondences, each of which has the same pose
  - Vote on pose, in an accumulator array
  - This is a Hough transform, with all its issues

THOMPSON & MUNDY'S ALGORITHM

- LOCAL FEATURE: VERTEX-PAIR



EACH VERTEX = INTERSECTION of 2 EDGES

GIVEN IMAGE w/ n vertices,
⇒ $2n^2$ vertex-pairs

- GIVEN ~~AN~~ A CORRESPONDENCE BETWEEN MODEL & IMAGE VERTICES & EDGES, THERE IS A UNIQUE TRANSFORMATION (AFFINE) WHICH MAPS IMAGE VERTEX-PAIR TO MODEL VERTEX-PAIR (⇒ MODELS WEAK PERSPECTIVE PROJECTION)

ALGORITHM

```
FOREACH IMAGE VERTEX-PAIR DO
  FOREACH MODEL VERTEX-PAIR DO
    COMPUTE 6-ELEMENT TRANSFORMATION
      T THAT MAPS ONE TO OTHER
    VOTE (INCREMENT "BIN") for 1
      POINT IN 6D TRANSFORM PARAM.
      SPACE CORRESPONDING TO T.
FIND "CLUSTERS" of VOTES (PEAK FINDING)
VERIFY PROJECTED MODEL w/ IMAGE
```

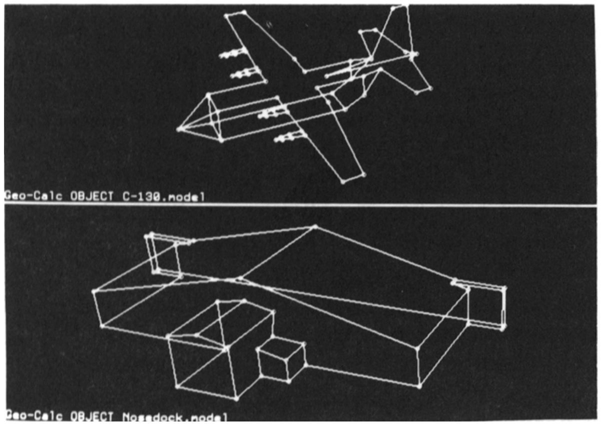
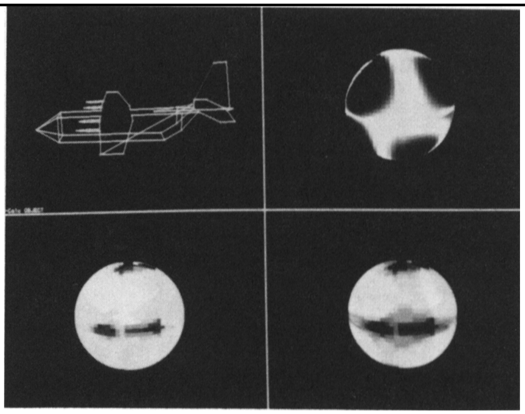


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE



From "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. ICCV, 1990

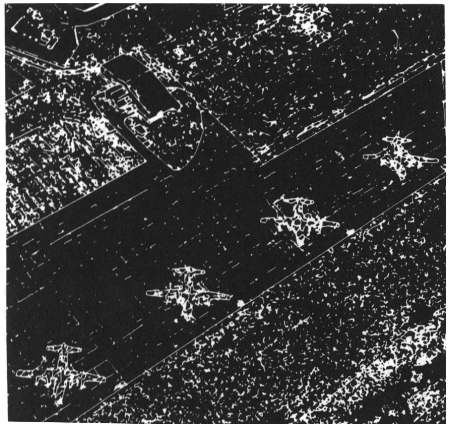


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990

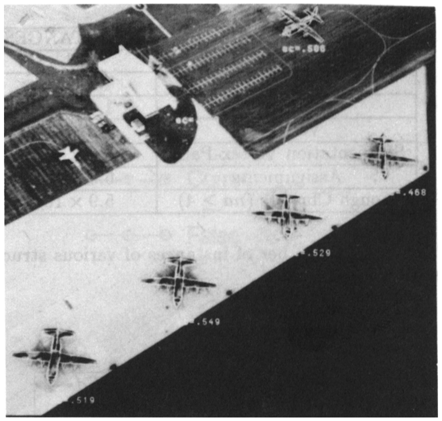


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990

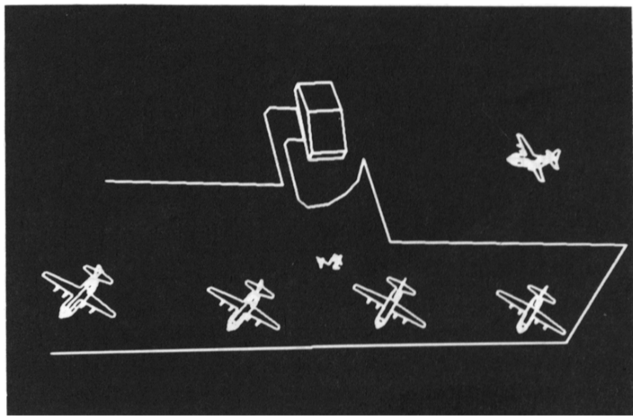


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990

GEOMETRIC HASHING

- Wolfson, Lamdan et al. 1988
- PLANAR, RIGID objects
- Affine camera model
- Property: An affine transform of a planar rigid body is uniquely defined by the transformation of 3 ordered, non-collinear pts [e.g., klein, 1925]

⇒ Pick 3 non-collinear model pts B_0, B_1, B_2 as affine basis

Any pt p represented by the affine coords (α, β) where

$$p = \alpha(B_1 - B_0) + \beta(B_2 - B_0) + B_0$$

(α, β) invariant under affine transf. T

$$Tp = \alpha(TB_1 - TB_0) + \beta(TB_2 - TB_0) + TB_0$$

Invariants

- There are geometric properties that are invariant to camera transformations
- Easiest case: view a plane object in weak perspective
- Assume we have three base points P_i on the object
  - then any other point on the object can be written as

$$P_k = P_1 + \mu_{ka}(P_2 - P_1) + \mu_{kb}(P_3 - P_1)$$

- Now image points are obtained by multiplying by a plane affine transformation, so

$$\begin{aligned} p_k &= AP_k \\ &= A\left(P_1 + \mu_{ka}(P_2 - P_1) + \mu_{kb}(P_3 - P_1)\right) \\ &= p_1 + \mu_{ka}(p_2 - p_1) + \mu_{kb}(p_3 - p_1) \end{aligned}$$

Invariants

- This means that, if I know the base points in the image, I can read off the µ values for the object
  - they're the same in object and in image --- **invariant**
- Suggests a strategy rather like the Hough transform
  - search correspondences, form µ's and vote

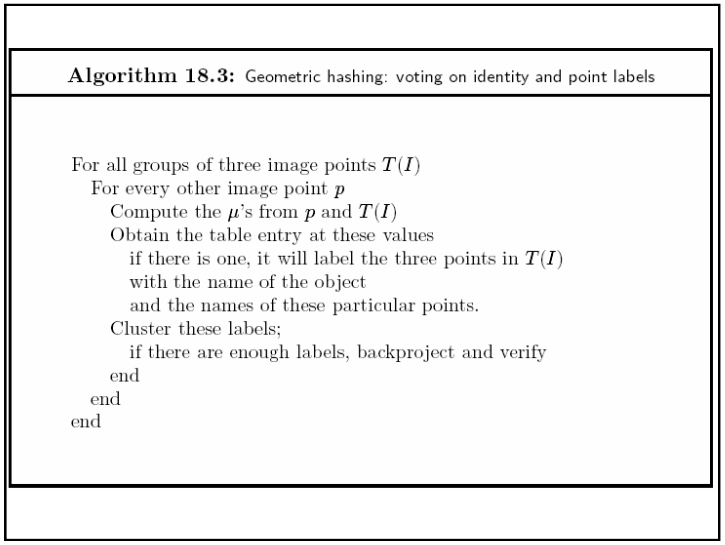
Geometric Hashing

- Vote on identity and correspondence using invariants
  - Take hypotheses with large enough votes
- Fill up a table, indexed by µ's, with
  - the base points and fourth point that yield those µ's
  - the object identity

Offline Modeling/Learning Phase

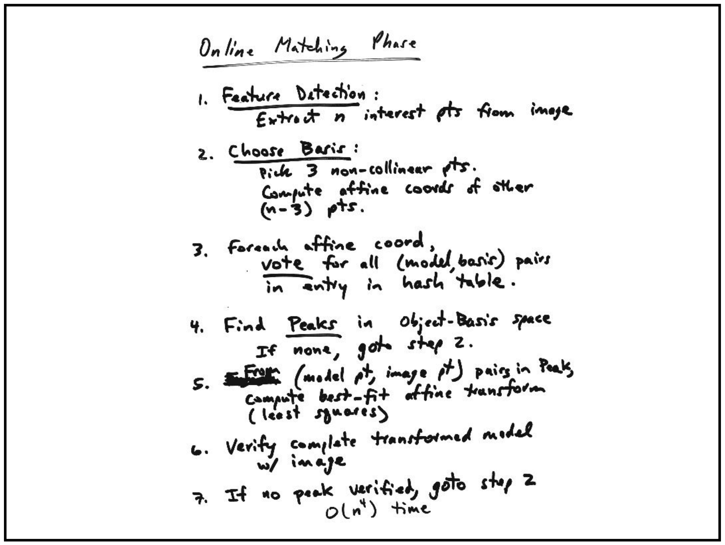
```
1. Select $m$ feature pts on planar
      model(s).
2. foreach triple ordered, non-collinear
   triple of model pts do
       Compute coords of other $(m-3)$ pts
       foreach coord $(\alpha,\beta)$ store
           Hash $(\alpha,\beta)$ = (object, basis)
```

$O(m^4)$ time and space.



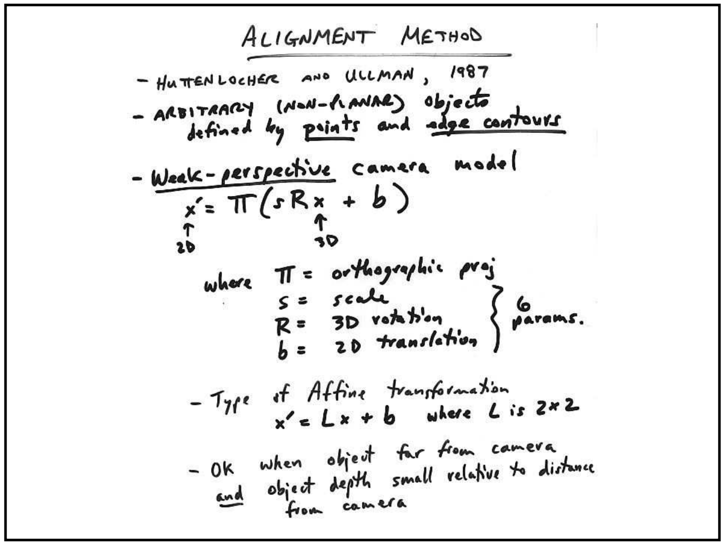
Algorithm 18.3: Geometric hashing: voting on identity and point labels

```
For all groups of three image points $T(I)$
  For every other image point $p$
    Compute the $\mu$'s from $p$ and $T(I)$
    Obtain the table entry at these values
      if there is one, it will label the three points in $T(I)$
      with the name of the object
      and the names of these particular points.
    Cluster these labels;
      if there are enough labels, backproject and verify
    end
  end
end
```



Online Matching Phase

1. Feature Detection:
   Extract n interest pts from image
2. Choose Basis:
   Pick 3 non-collinear pts.
   Compute affine coords of other $(n-3)$ pts.
3. Foreach affine coord,
   vote for all (model, basis) pairs
   in entry in hash table.
4. Find Peaks in Object-Basis space
   If none, goto step 2.
5. From (model pt, image pt) pairs in Peaks,
   Compute best-fit affine transform
   (least squares)
6. Verify complete transformed model
   w/ image
7. If no peak verified, goto step 2
   $O(n^4)$ time



ALIGNMENT METHOD

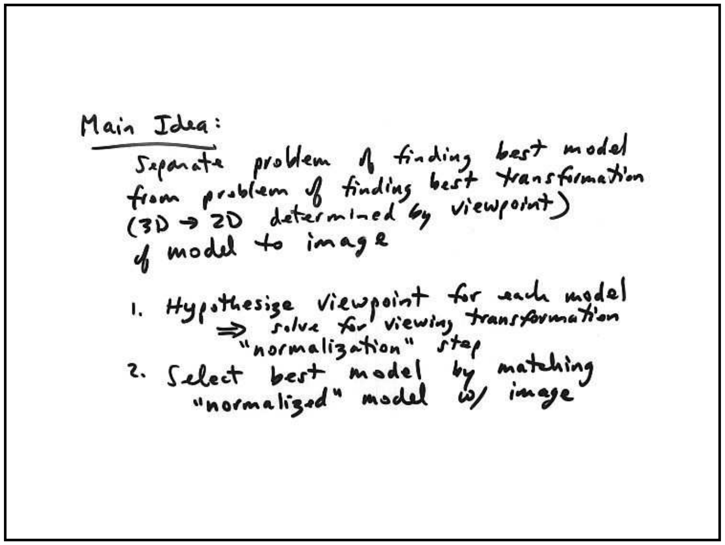
- HUTTENLOCHER AND ULLMAN, 1987
- ARBITRARY (NON-PLANAR) objects
  defined by points and edge contours
- Week-perspective camera model

$$\underset{\substack{\uparrow \\ 2D}}{x'} = \Pi(sR\underset{\substack{\uparrow \\ 3D}}{x} + b)$$

where Π = orthographic proj

$\left.\begin{array}{l} s = \text{scale} \\ R = \text{3D rotation} \\ b = \text{2D translation} \end{array}\right\}$ 6 params.

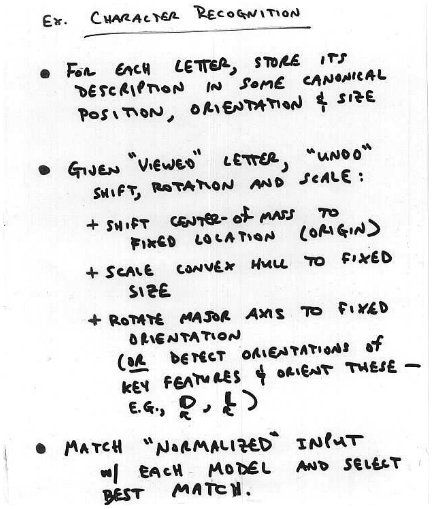
- Type of Affine transformation
  $x' = Lx + b$ where L is 2×2
- OK when object far from camera
  and object depth small relative to distance
  from camera



Main Idea:

Separate problem of finding best model from problem of finding best transformation ($3D \rightarrow 2D$ determined by viewpoint) of model to image

1. Hypothesize viewpoint for each model
   ⇒ solve for viewing transformation
   "normalization" step
2. Select best model by matching
   "normalized" model w/ image



EX. <u>CHARACTER RECOGNITION</u>

- FOR EACH LETTER, STORE ITS DESCRIPTION IN SOME CANONICAL POSITION, ORIENTATION & SIZE
- GIVEN "VIEWED" LETTER, "UNDO" SHIFT, ROTATION AND SCALE:
  + SHIFT CENTER-OF MASS TO FIXED LOCATION (ORIGIN)
  + SCALE CONVEX HULL TO FIXED SIZE
  + ROTATE MAJOR AXIS TO FIXED ORIENTATION (<u>OR</u> DETECT ORIENTATIONS OF KEY FEATURES & ORIENT THESE — E.G., $\underset{R}{D}$, $\underset{R}{L}$)
- MATCH "NORMALIZED" INPUT W/ EACH MODEL AND SELECT BEST MATCH.

Recognizing Human Actions

- Movement and posture change
  - run, walk, crawl, jump, hop, swim, skate, sit, stand, kneel, lie, dance (various), …
- Object manipulation
  - pick, carry, hold, lift, throw, catch, push, pull, write, type, touch, hit, press, stroke, shake, stir, turn, eat, drink, cut, stab, kick, point, drive, bike, insert, extract, juggle, play musical instrument (various)…
- Conversational gesture
  - point, …
- Sign Language

Activities and Situation Assessment

- Example: Withdrawing money from an ATM
- Activities constructed by composing actions. Partial order plans may be a good model.
- Activities may involve multiple agents
- Detecting unusual situations or activity patterns is facilitated by the video activity transform

Objects in Space

- Segment/Region-of-interest
- Features (points, curves, wavelet coefficients..)
- Correspondence and deform into alignment
- Recover parameters of generative model
- Discriminative classifier

Actions in Space-Time

- Segment/volume-of-interest
- Features (points, curves, wavelets, motion vectors..)
- Correspondence and deform into alignment
- Recover parameters of generative model
- Discriminative classifier

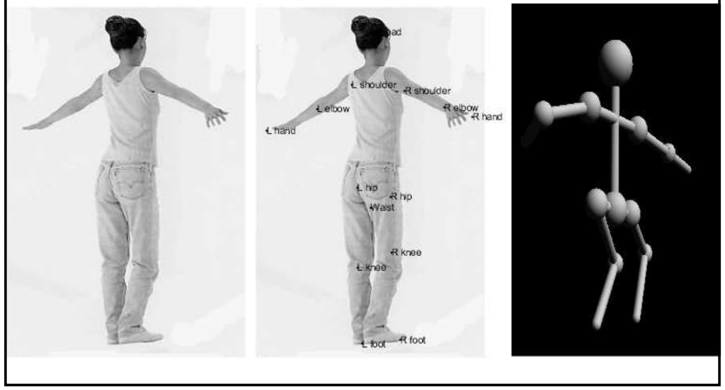
Key Cues for Action Recognition

- “Morpho-kinesics” of action (shape and movement of the body)
- Identity of the object/s
- Activity context

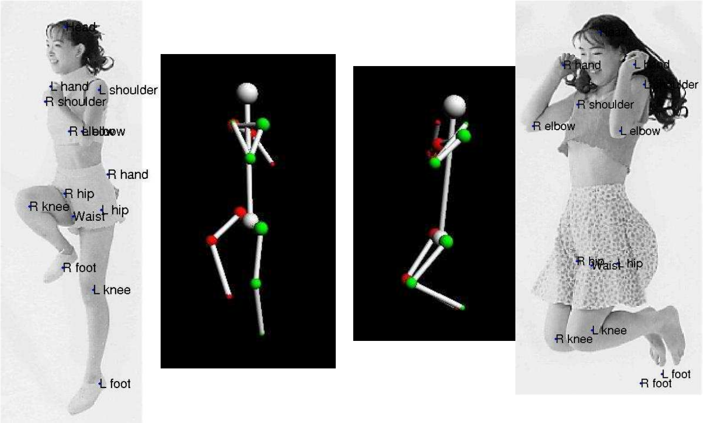
Image/Video Stick figure Action

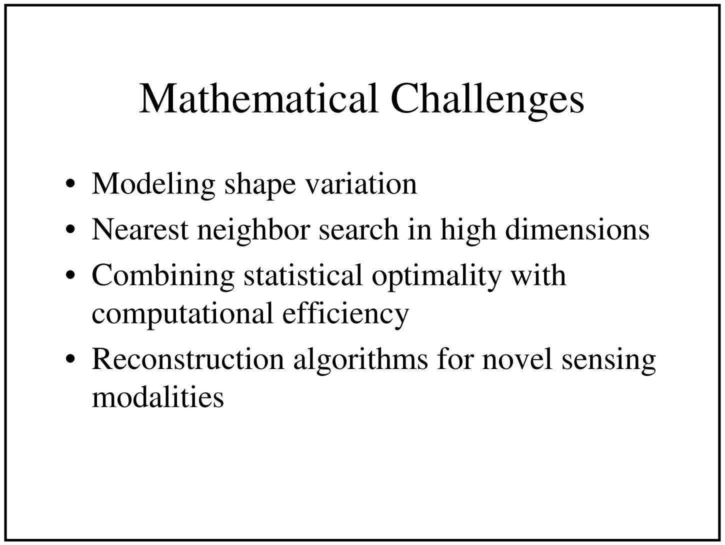
- Stick figures can be specified in a variety of ways or at various resolutions (degrees of freedom)
  - 2D joint positions
  - 3D joint positions
  - Joint angles
- Complete representation
- Evidence that it is effectively computable

Human Body Configurations



Human Body Configurations





Mathematical Challenges

- Modeling shape variation
- Nearest neighbor search in high dimensions
- Combining statistical optimality with computational efficiency
- Reconstruction algorithms for novel sensing modalities