COMPUTER VISION AND OBJECT RECOGNITION

Motivation: For a robot to be truly autonomous, it must sense the world directly.

Computer Vision: "Field of AI that describes the automatic understanding of the structure and properties of a possibly changing 3D world from its 2D images" [DAA95]

Set of processes that recover descriptions of the world from images.

```
OBJECT
  |     camera records light and
  |     maps it to a grid of cells
  V
IMAGE PLANE
  |
  |     SAMPLING: selecting a discrete set of
  |     locations on the plane
  |     &
  |     QUANTIZATION: assigning to each selected
  |     location an integer that represents the
  |     irradiance (brightness) of the point
  V
DIGITAL IMAGE
```

DIGITAL IMAGE

(Taken from Dean, Allen, & Aloimonos' AI textbook)

a.

183	196	199	200	214	215	118	226	98	104
208	194	200	226	157	88	76	157	0	43
209	214	199	182	91	71	59	173	217	177
214	214	175	150	88	71	59	138	217	214
193	215	208	199	113	60	55	52	244	199
138	105	137	152	215	109	71	44	70	168
137	120	105	102	104	157	244	137	75	68
140	123	120	123	105	105	120	137	244	199
138	118	139	109	108	138	138	138	138	168
109	114	121	121	138	119	119	138	138	152

b.

COMPUTER VISION

Need for Vision/Perception:

1. Manipulation: e.g. grasping an object
2. Navigation: e.g. finding clear paths, avoiding obstacles
3. Object Recognition: e.g. distinguishing between friendly and unfriendly objects.

Applications:

- Aerial Photo Interpretation

  Computers are been trained to detect changes in vegetation, movement of military equipment, etc.

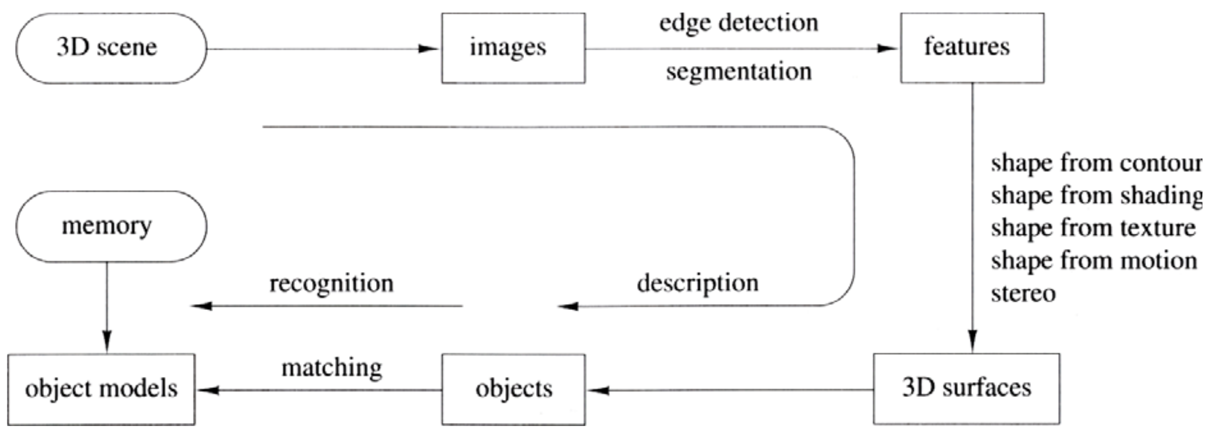
- Object Recognition: e.g. Face recognition:
  - Security systems
  - Teleconferencing:

    new poses and facial expressions are synthesized from existing images.

EXTRACTING 3D INFORMATION USING VISION

1. Detemining Objects: Segmentation
2. Determining Pose: position and orientation
3. Determining Shape

(Taken from Dean, Allen, & Aloimonos' AI textbook)



EXTRACTING 3D INFORMATION USING VISION

1. **Segmentation:**
   - Segmentation of the scene into distinct objects
   - Organizing the array of image pixels into regions that would correspond to semantically meaningful entities in the scene.
   - One technique: edge detection using line labeling (studied before)

2. **Determining Position and Orientation** of each object w.r.t. observer

   (Important for navigation and manipulation)

   - Position of a point P = (x, y, z) Orientation =
     - of the object as a whole:

       specified in terms of a 3D rotation relating its coordinate frame to that of the camera

     - of the surface of the object at point P:

       specified by a unit normal vector.

EXTRACTING 3D INFORMATION USING VISION

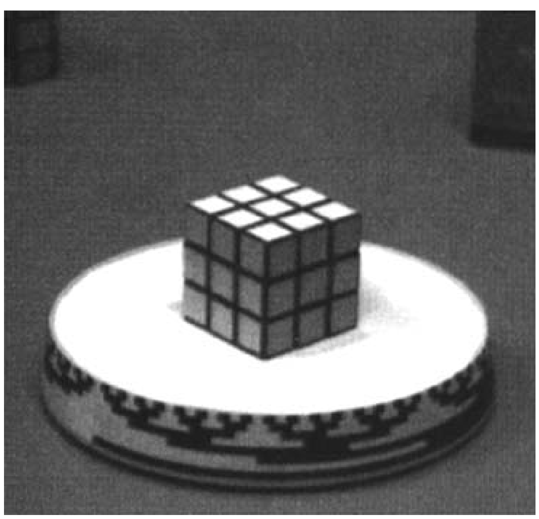
(Figures in this section taken from Russell and Norvig's AI textbook)

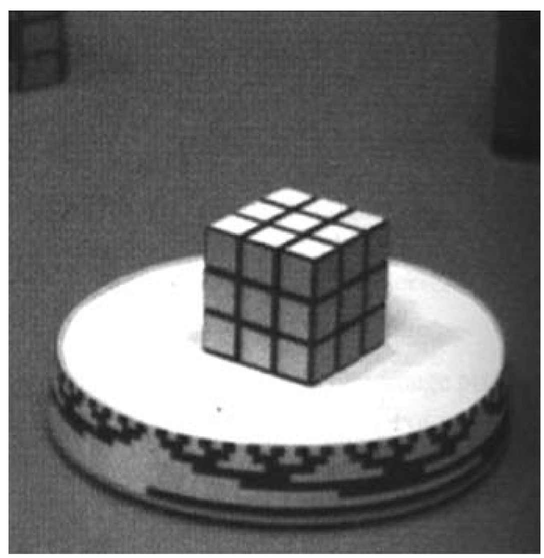
3. **DETERMINING SHAPE:** of each object

   (Important for manipulation and, together with color and texture, very important for object recognition)

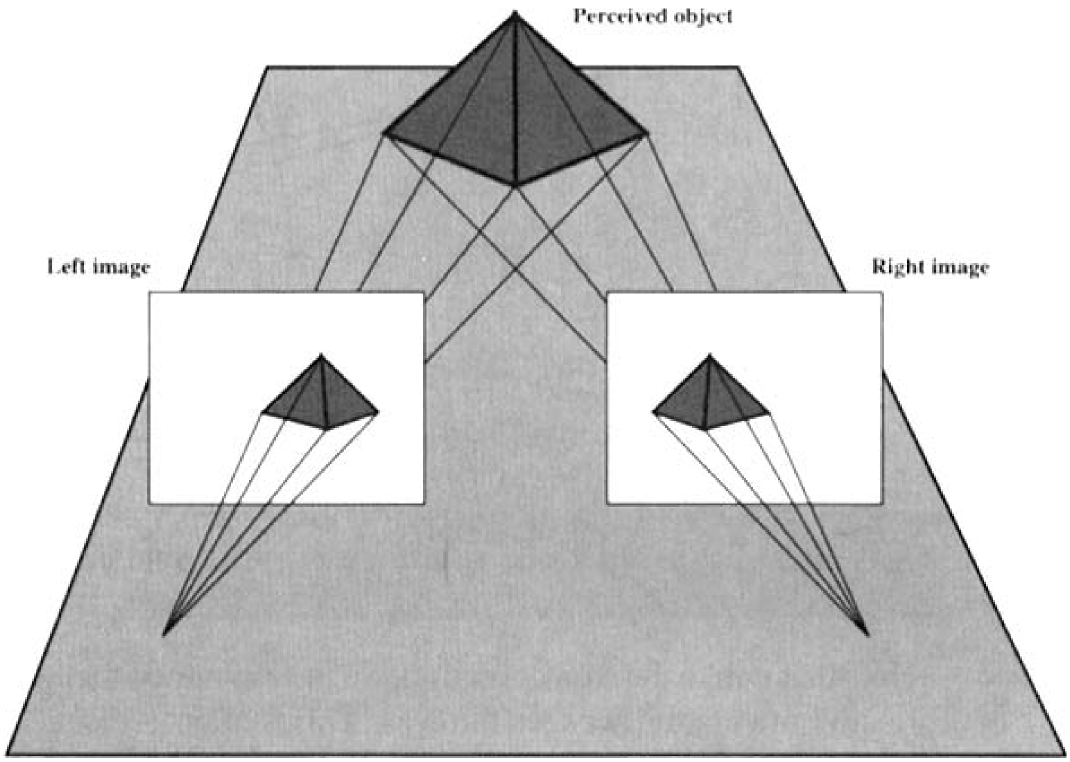
   Shape is what remains unchanged by rotations and translation (e.g. cylinders, cones, spheres, ...)

   (a) Determining Shape from Motion
   Two images separated by time



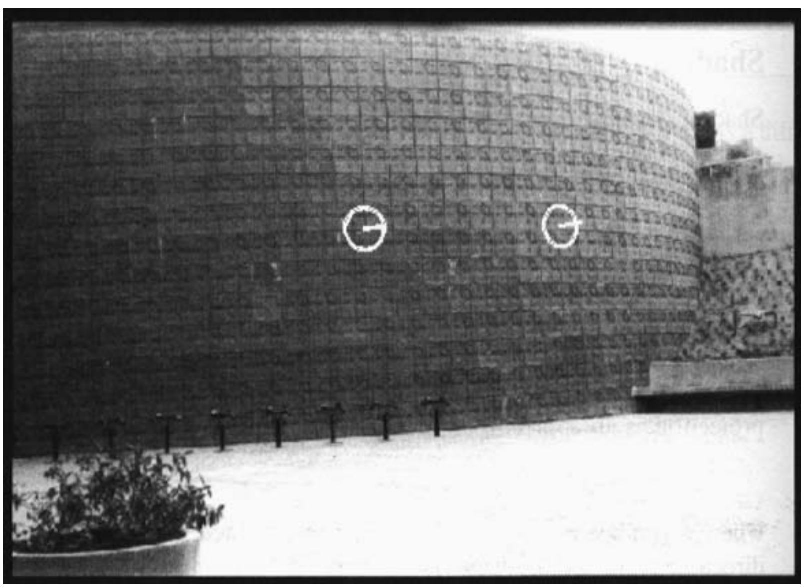


(b) Shape from Binocular stereopsis
Two images separated by space



(c) Shape from Texture

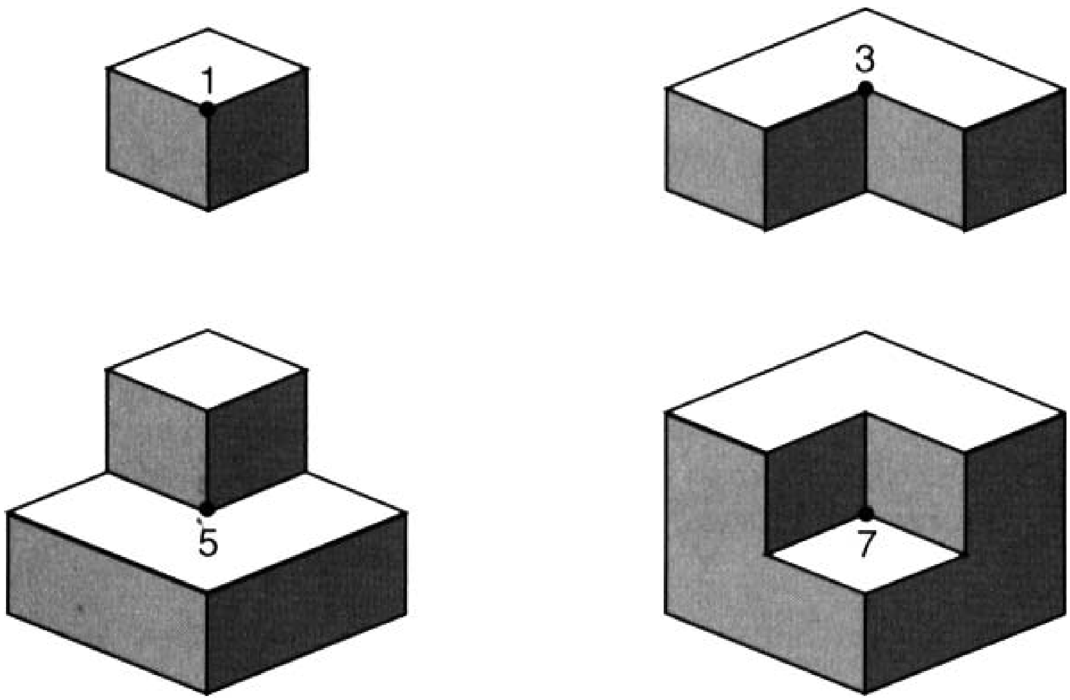
Texture is a tactile quality of an object



(d) Shape from Shading

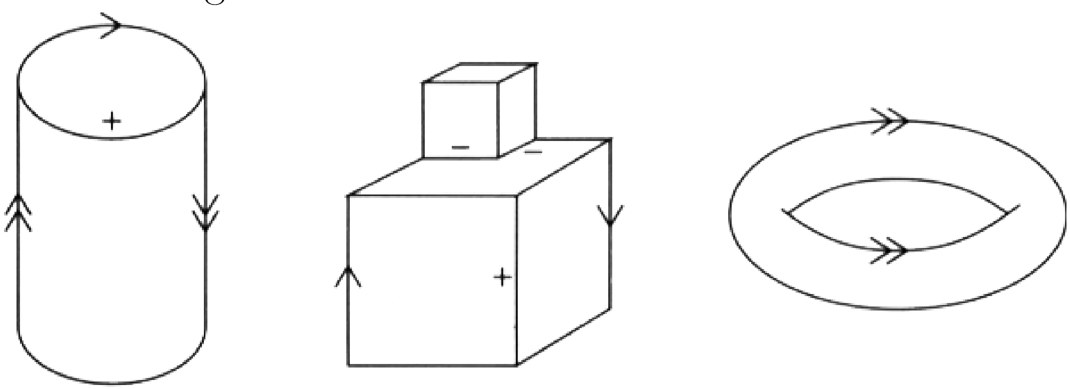
Variation on the intensity of light (brightness)

(See Winston - Chapter 27)



(e) Shape from Contour

line labeling



RECOGNIZING OBJECTS

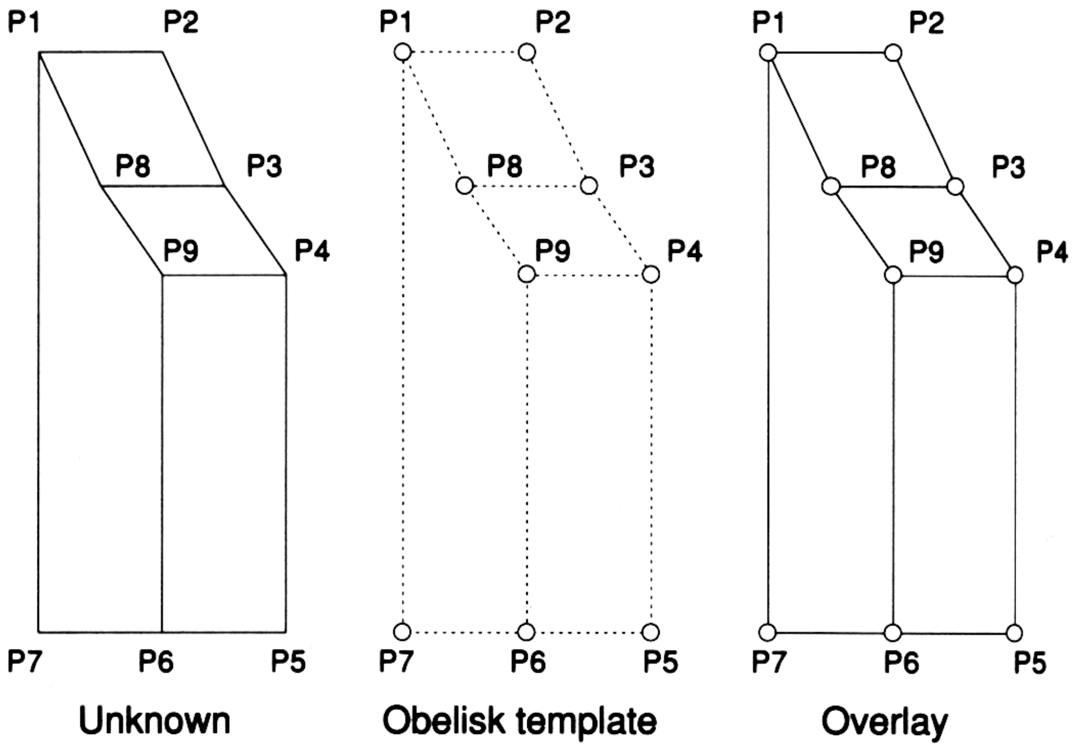
(Figures in this section are taken from Winston's AI textbook)

Determine shape by matching the object against templates

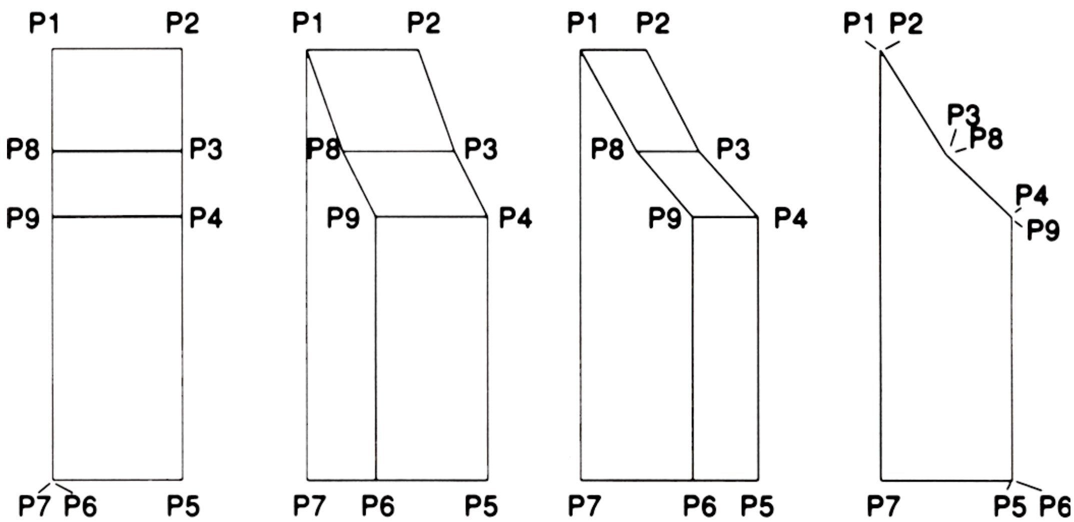
- The object is represented by a set of 3D points $P_1, \ldots, P_n$ measured in a coordinate system natural for the object
- Make knowledge of shape explicit by deriving 3D coordinate values for all vertexes
- Assumptions:
  - Solid objects
  - Orthographic projections: $(x, y, z) \longrightarrow (x, y)$

SAME ORIENTATION

Just Match



ALLOWING ROTATIONS AROUND $y - axis$



ALLOWING ROTATIONS AROUND $y - axis$

Note that after rotating θ:

- x changes,
  $x_\theta = x \cos\theta - z \sin\theta$
- y does not change,
  $y_\theta = y$
- z changes, but after the projection it disappears $z_\theta = x \sin\theta + z \cos\theta$

ALLOWING ROTATIONS AROUND $y - axis$

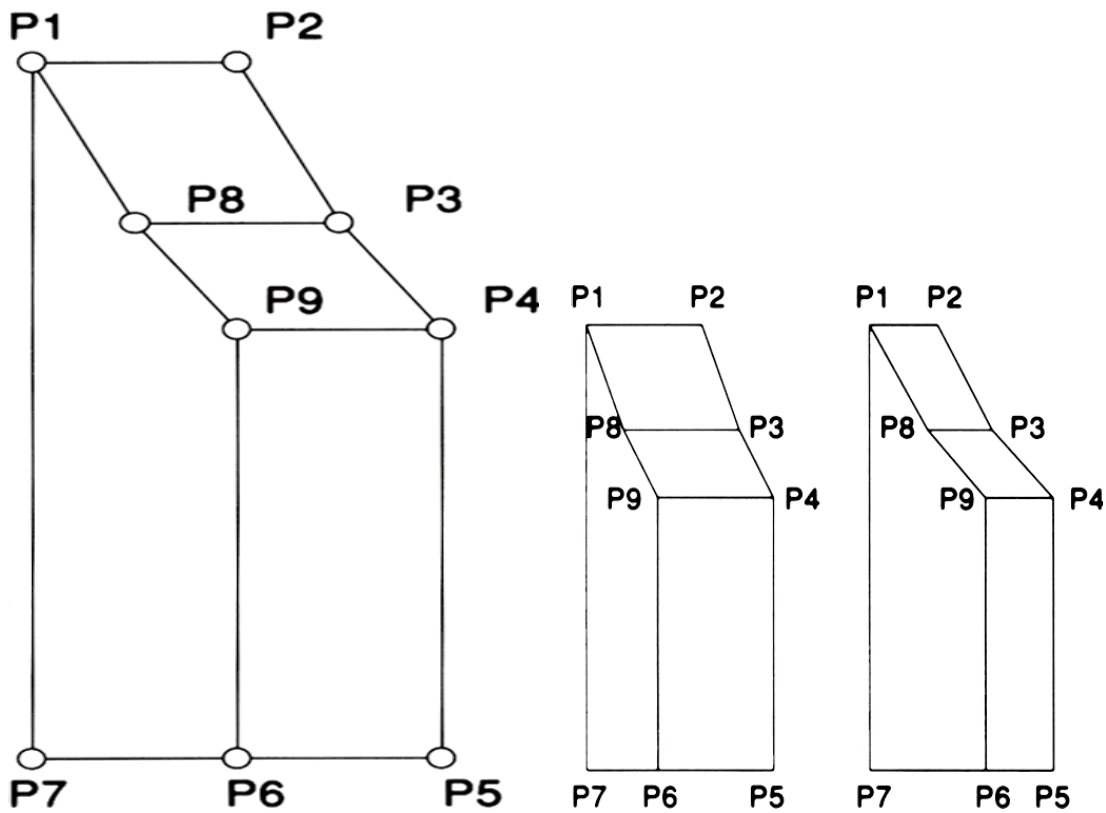
For matching against template, we need:

- 2 model templates
- to find α and β for x.

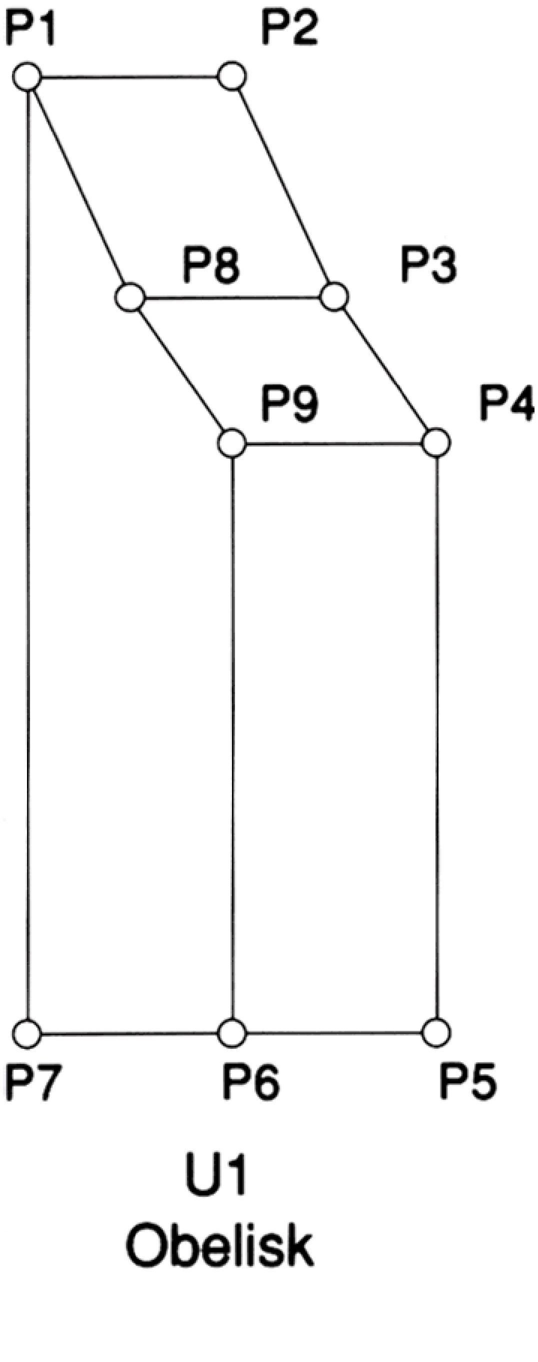
$$x_{I_0} = \alpha x_{I_1} + \beta x_{I_2}$$

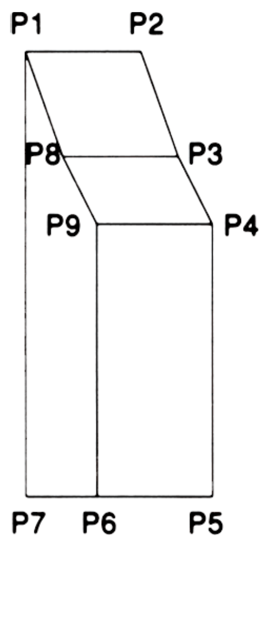
- 2 corresponding points:

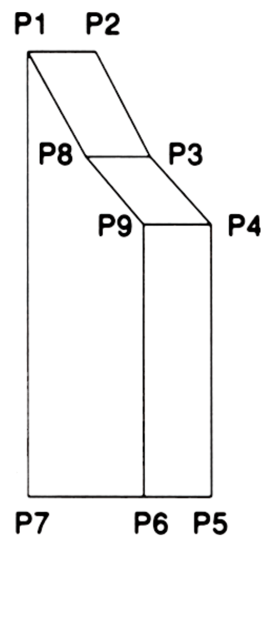
  because we need two equations to find α and β.



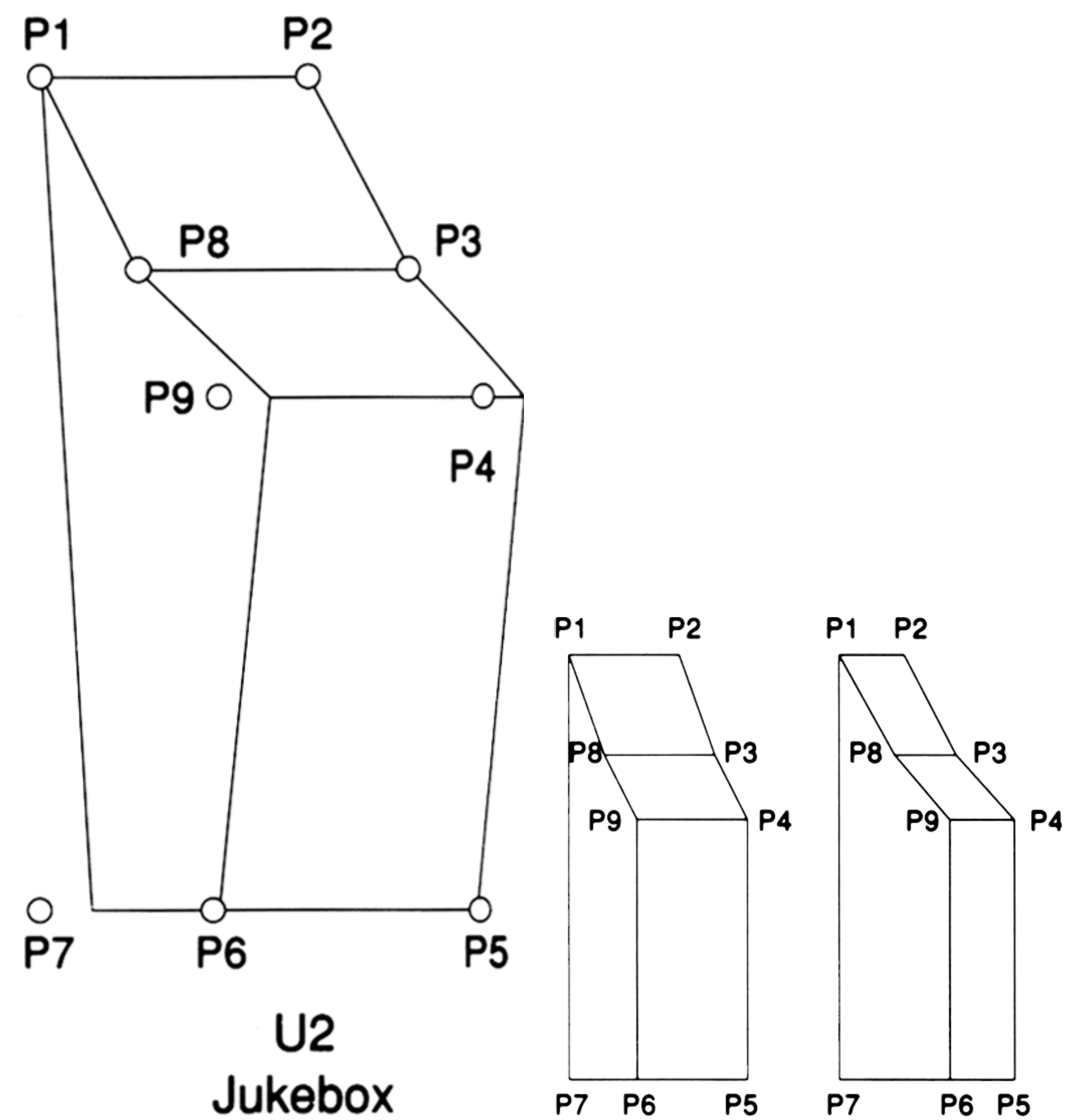
EXAMPLE 1 - The object fits the templates







EXAMPLE 2 - The object does not fit the templates



ALLOWING GENERAL ROTATIONS AND TRANSLATIONS

Allowed: Rotations around all 3 axes and translations.

$$x_\theta = ax + by + cz + d$$

where a, b, c depend on the rotation angles and d depends on the translation.

For this case, we need:

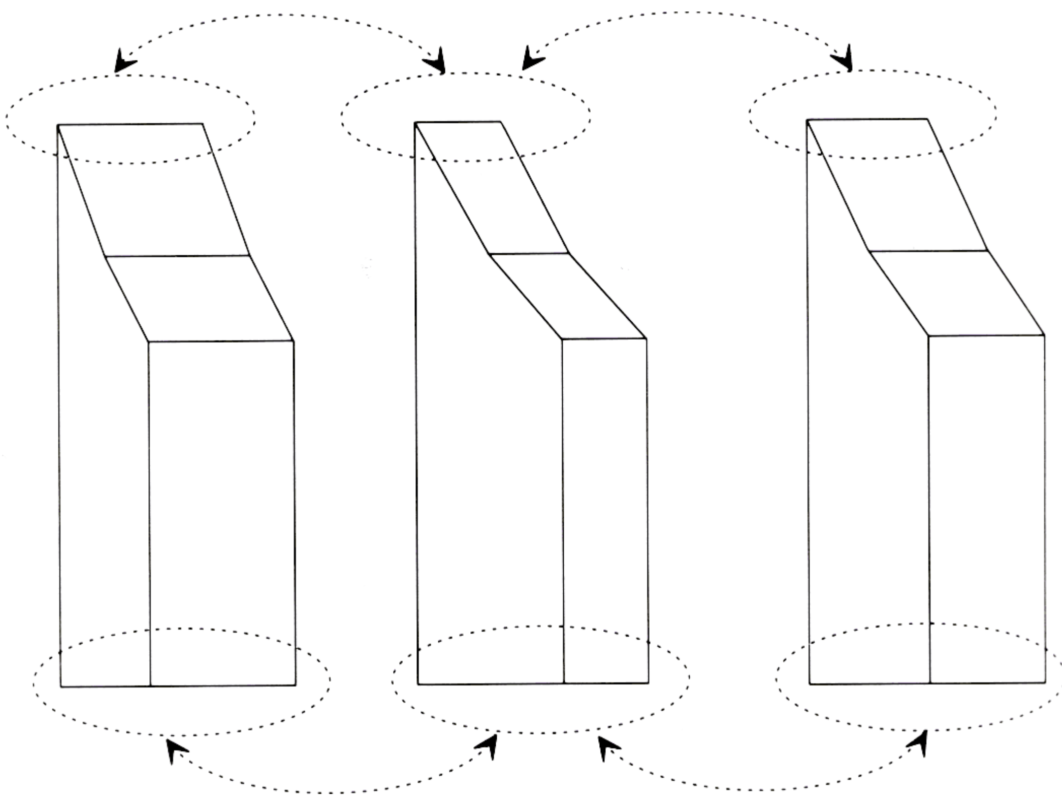
- 3 image templates

$$x_{I_0} = \alpha_x x_{I_1} + \beta_x x_{I_2} + \gamma_x x_{I_3} + \delta_x$$

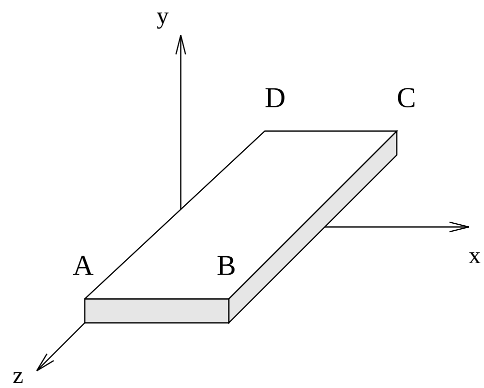
- 4 corresponding points
- to find $\alpha_x, \beta_x, \gamma_x, \delta_x$ for x
- to find $\alpha_y, \beta_y, \gamma_y, \delta_y$ for y

FINDING CORRESPONDING POINTS

- Difficult
- No need for (one-to-one) corresponding points, just corresponding sets of points.
- Use heuristics to find corresponding sets. (e.g. natural top or bottom, if any)



EXAMPLE



Consider the thin rectangle I_1 shown in the figure. Let I_2 be the same thin rectangle rotated by a certain angle around the y–axis. Using I_1 and I_2 as templates, determine whether or not the unknown object I_0 is the same as the template rectangles (up to a rotation around the y–axis). In other words, determine if I_0 "fits" the templates.

The values of the x coordinates for the four corresponding vertices A, B, C, and D of the objects are given below. Remember that, for any given point P of the rectangle, the x coordinates of the corresponding points PI_0 (of the unknown object), PI_1 (of the template I_1), and PI_2 (of the template I_2), are related by the following linear equation:

$$x_{PI_0} = \alpha\ x_{PI_1} + \beta\ x_{PI_2}$$

Object	x coordinate of point A	x coordinate of point B	x coordinate of point C	x coordinate of point D
Template: I_1	0	1	1	0
Template: I_2	1	1	-2	-2
Unknown: I_0	0.8	1.4	-1	-1.8

Solution: From $x_{AI_0} = \alpha * x_{AI_1} + \beta * x_{AI_2}$ and $x_{BI_0} = \alpha * x_{BI_1} + \beta * x_{BI_2}$

we get $0.8 = \alpha * 0 + \beta * 1$ and $1.4 = \alpha * 1 + \beta * 1$

hence, $\alpha = 0.6$ and $\beta = 0.8$.

Therefore $x_{CI_0} = \alpha * x_{CI_1} + \beta * x_{CI_2}$ = 0.6 * 1 + 0.8 * (-2) = -1

and $x_{DI_0} = \alpha * x_{DI_1} + \beta * x_{DI_2}$ = 0.6 * 0 + 0.8 * (-2) = -1.6

Because the expected value of the x coordinate of point D of the unknown object (-1.6) does not match the actual value of that coordinate (-1.8), the unknown object does not fit the templates.