Vision

Chapter 24

Outline

- ♢ Perception generally
- ♢ Image formation
- ♢ Early vision
- ♢ 2D $\longrightarrow$ 3D
- ♢ Object recognition

Perception generally

Stimulus (percept) S, World W

$$S = g(W)$$

E.g., $g =$ "graphics." Can we do vision as inverse graphics?

$$W = g^{-1}(S)$$

Perception generally

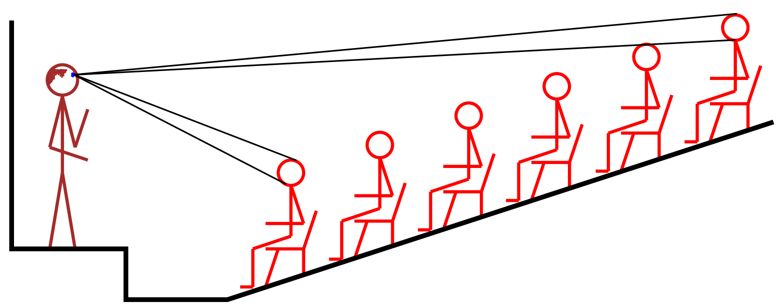
Stimulus (percept) S, World W

$$S = g(W)$$

E.g., $g =$ "graphics." Can we do vision as inverse graphics?

$$W = g^{-1}(S)$$

Problem: massive ambiguity!



Perception generally

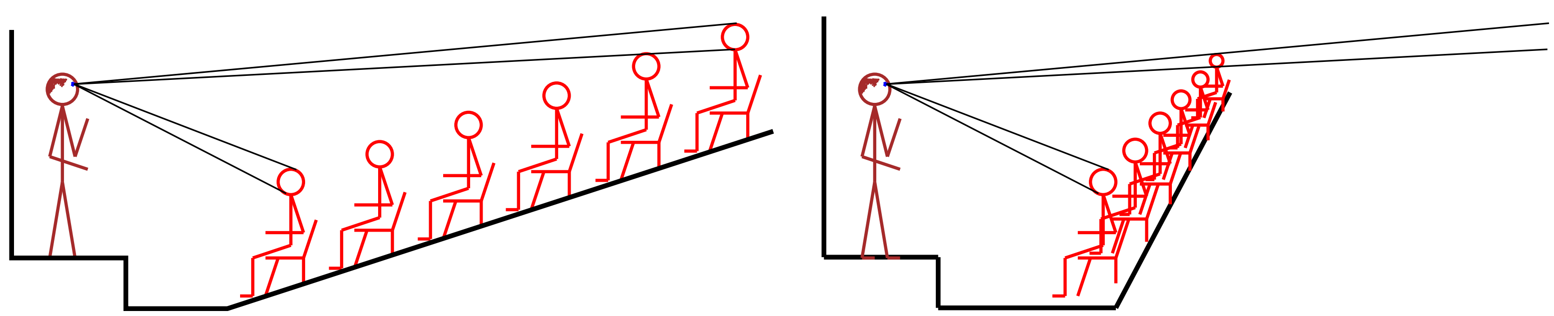
Stimulus (percept) S, World W

$$S = g(W)$$

E.g., $g =$ "graphics." Can we do vision as inverse graphics?

$$W = g^{-1}(S)$$

Problem: massive ambiguity!



Perception generally

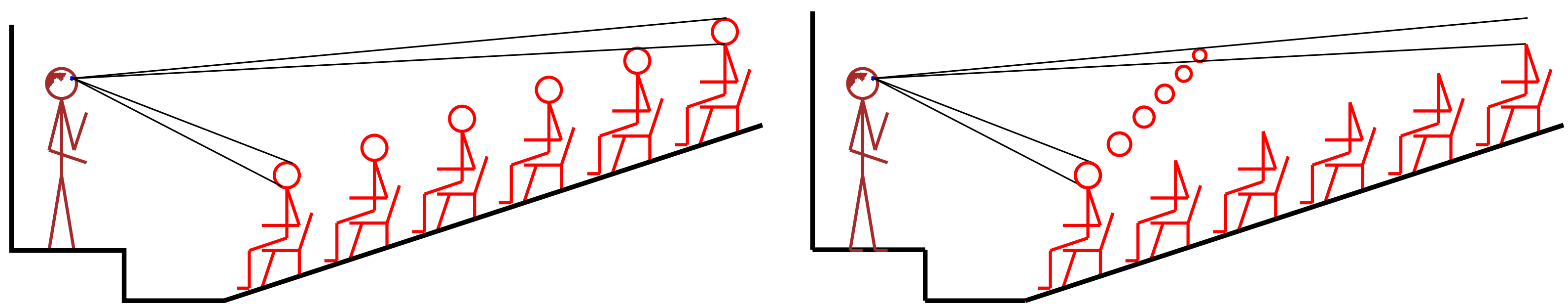
Stimulus (percept) S, World W

$$S = g(W)$$

E.g., $g =$ "graphics." Can we do vision as inverse graphics?

$$W = g^{-1}(S)$$

Problem: massive ambiguity!



Better approaches

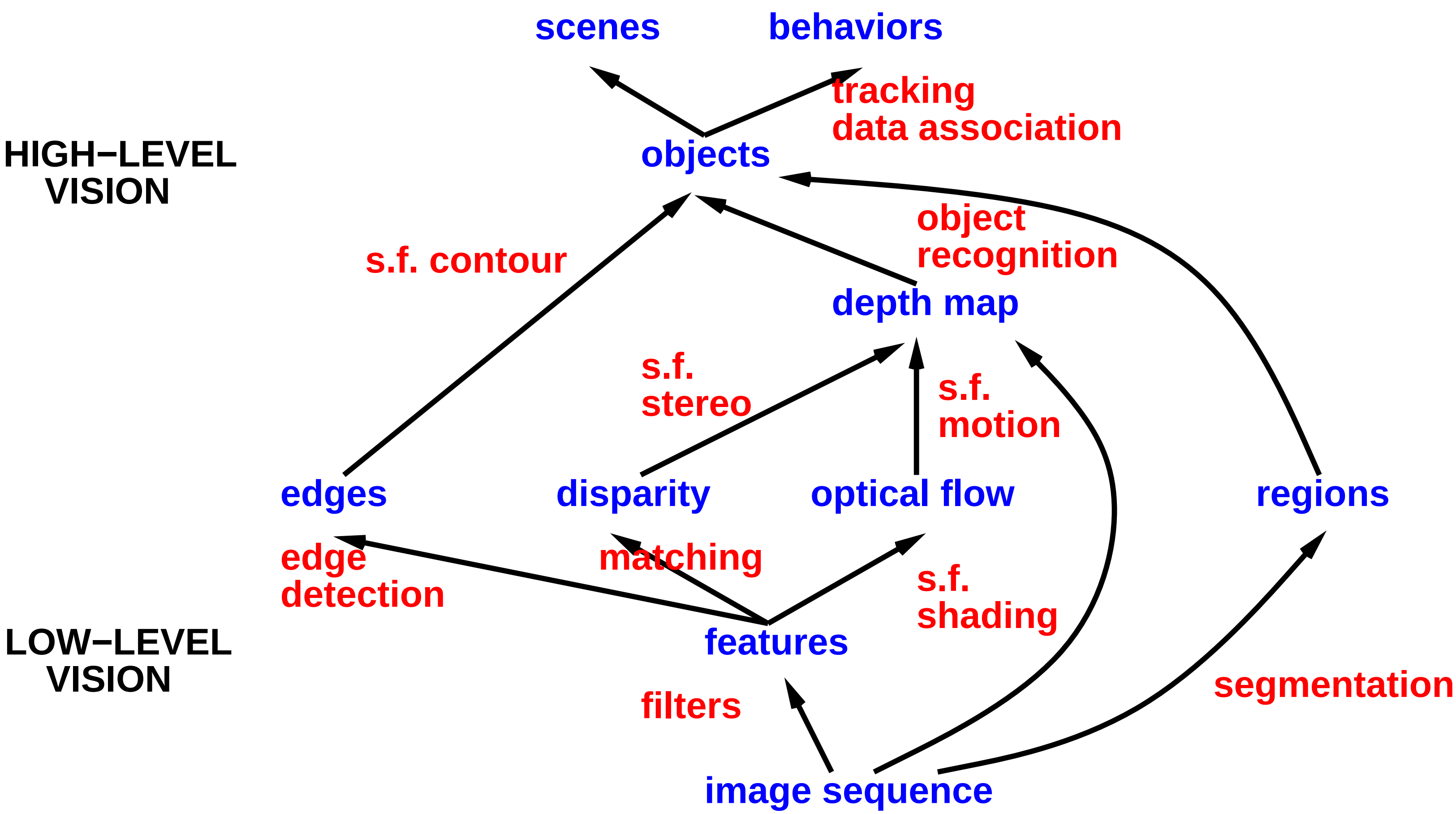
Bayesian inference of world configurations:

$$P(W|S) = \alpha \underbrace{P(S|W)}_{\text{“graphics”}} \underbrace{P(W)}_{\text{“prior knowledge”}}$$

Better still: no need to recover exact scene!
Just extract information needed for

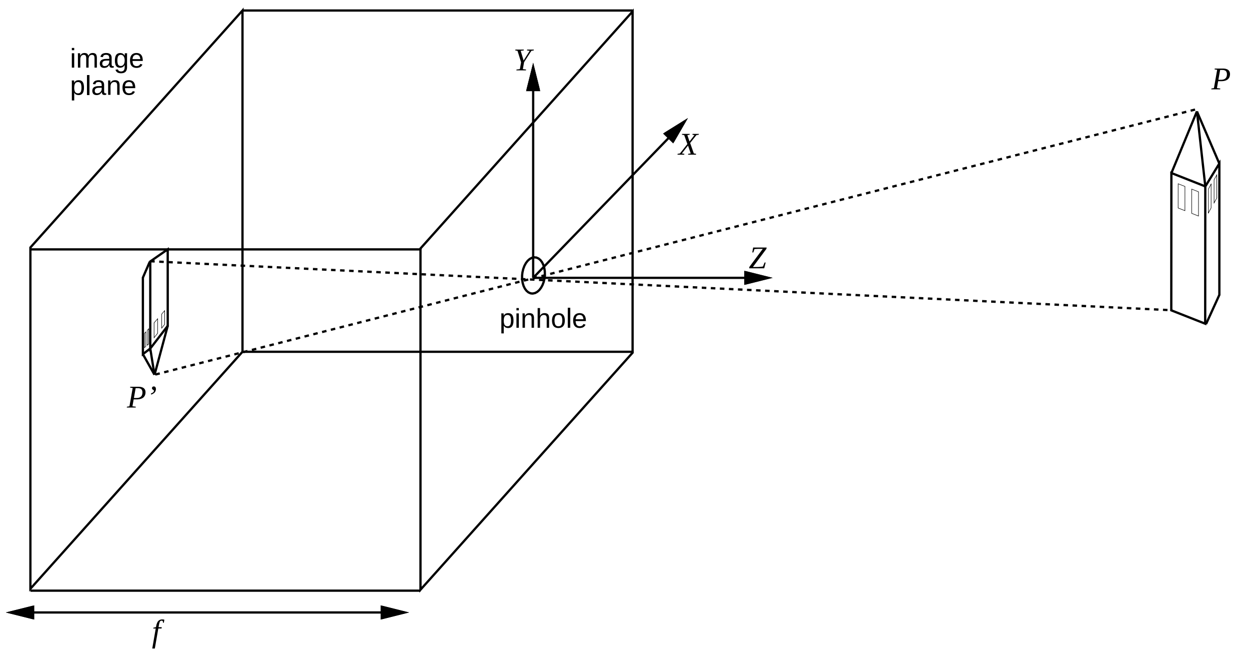
- navigation
- manipulation
- recognition/identification

Vision "subsystems"



Vision requires combining multiple cues

Image formation



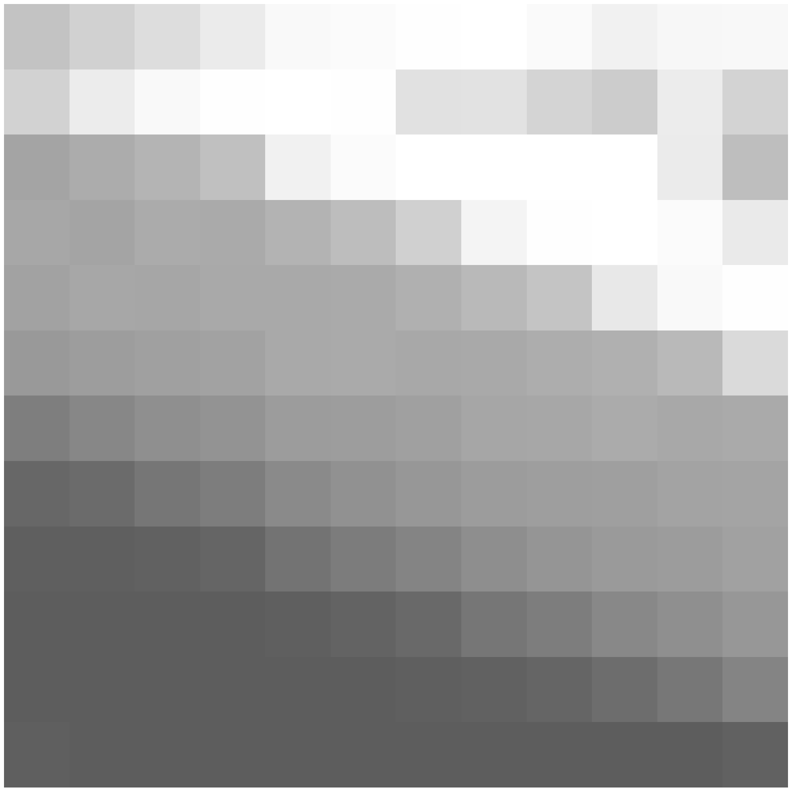
P is a point in the scene, with coordinates (X, Y, Z)
P' is its image on the image plane, with coordinates (x, y, z)

$$x = \frac{-fX}{Z}, \;\; y = \frac{-fY}{Z}$$

by similar triangles. Scale/distance is indeterminate!

Images

Images contd.



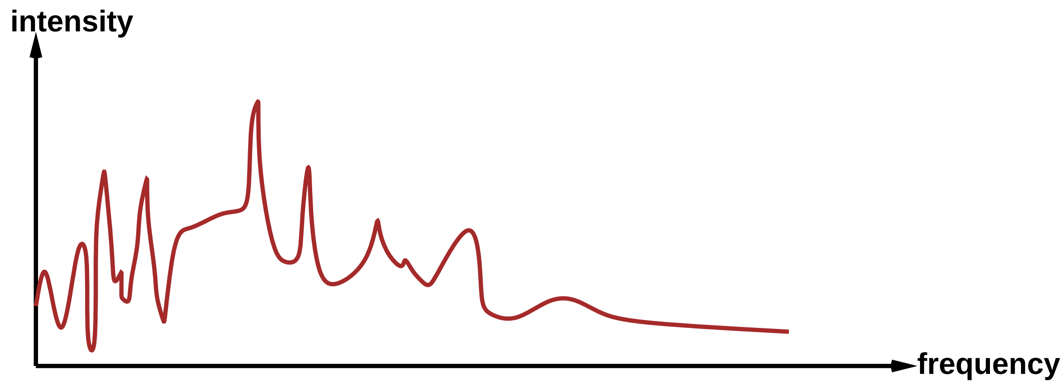
195	209	221	235	249	251	254	255	250	241	247	248
210	236	249	254	255	254	225	226	212	204	236	211
164	172	180	192	241	251	255	255	255	255	235	190
167	164	171	170	179	189	208	244	254	255	251	234
162	167	166	169	169	170	176	185	196	232	249	254
153	157	160	162	169	170	168	169	171	176	185	218
126	135	143	147	156	157	160	166	167	171	168	170
103	107	118	125	133	145	151	156	158	159	163	164
095	095	097	101	115	124	132	142	117	122	124	161
093	093	093	093	095	099	105	118	125	135	143	119
093	093	093	093	093	093	095	097	101	109	119	132
095	093	093	093	093	093	093	093	093	093	093	119

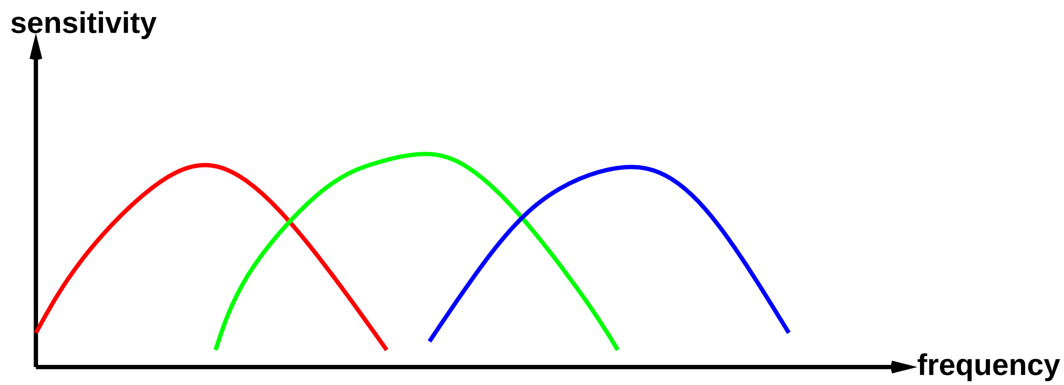
$I(x, y, t)$ is the intensity at (x, y) at time t

CCD camera $\approx$ 1,000,000 pixels; human eyes $\approx$ 240,000,000 pixels
i.e., 0.25 terabits/sec

Color vision

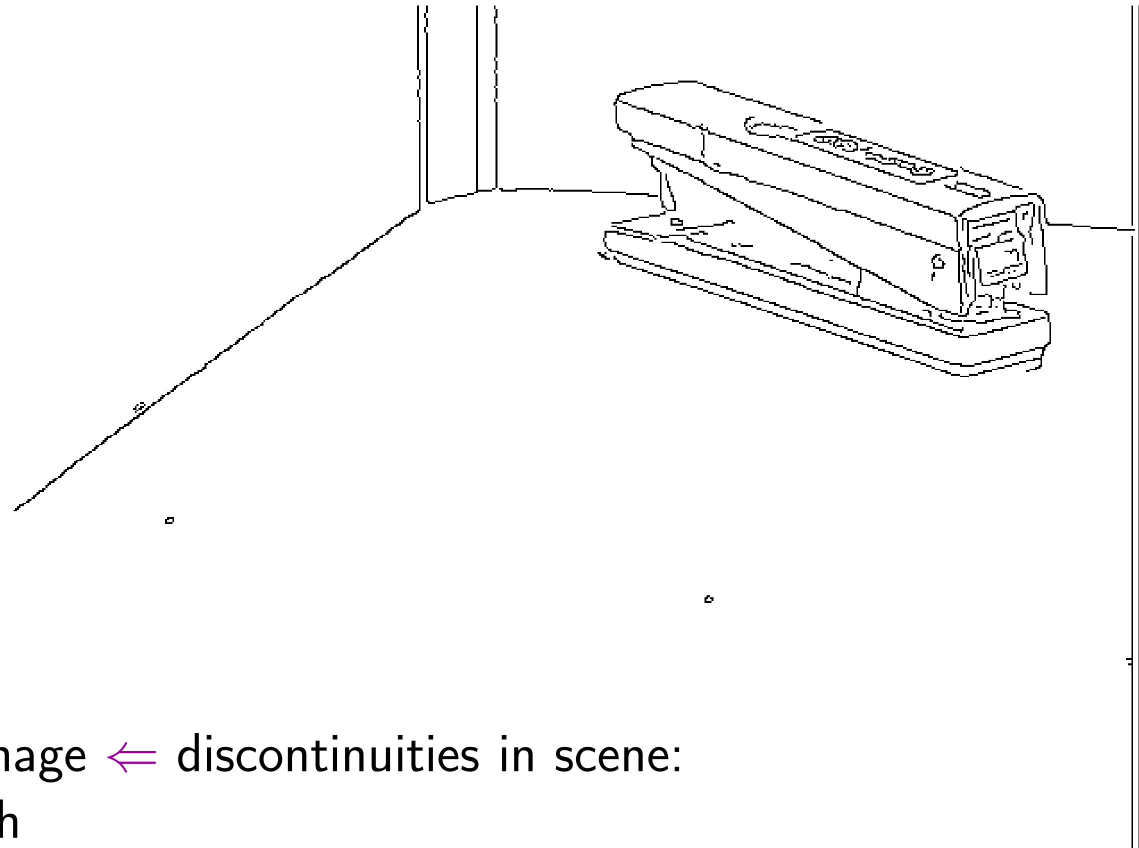
Intensity varies with frequency → infinite-dimensional signal





Human eye has three types of color-sensitive cells;
each integrates the signal ⇒ 3-element vector intensity

Edge detection



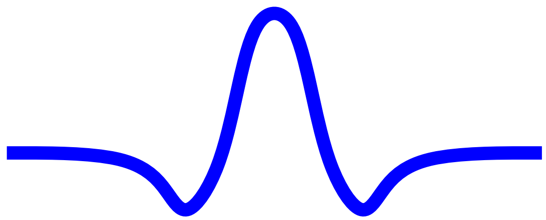
Edges in image $\Leftarrow$ discontinuities in scene:

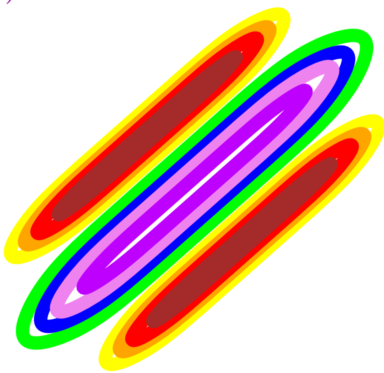
1) depth
2) surface orientation
3) reflectance (surface markings)
4) illumination (shadows, etc.)

Edge detection contd.

1) Convolve image with spatially oriented filters (possibly multi-scale)

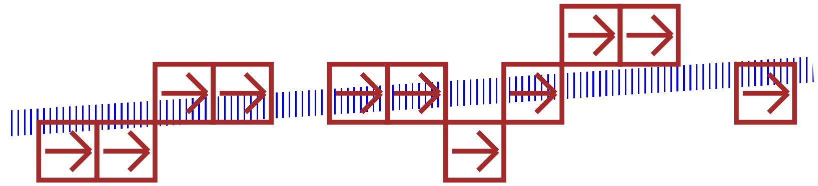
$$E_\theta(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_\theta(u, v) I(x + u, y + v)\, du\, dv$$





2) Label above-threshold pixels with edge orientation

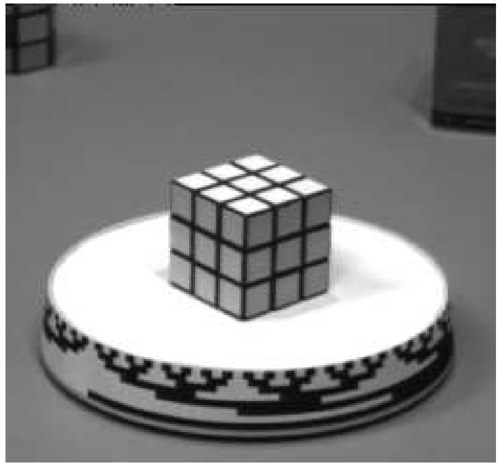
3) Infer "clean" line segments by combining edge pixels with same orientation

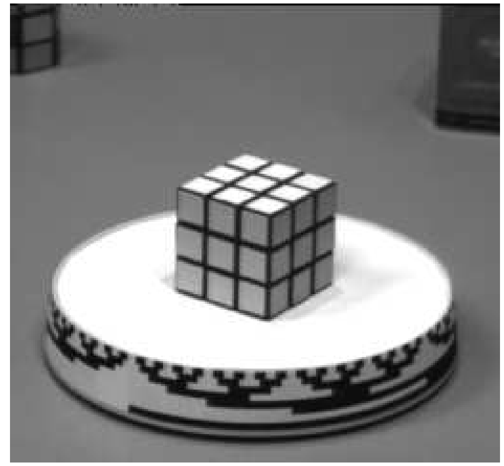


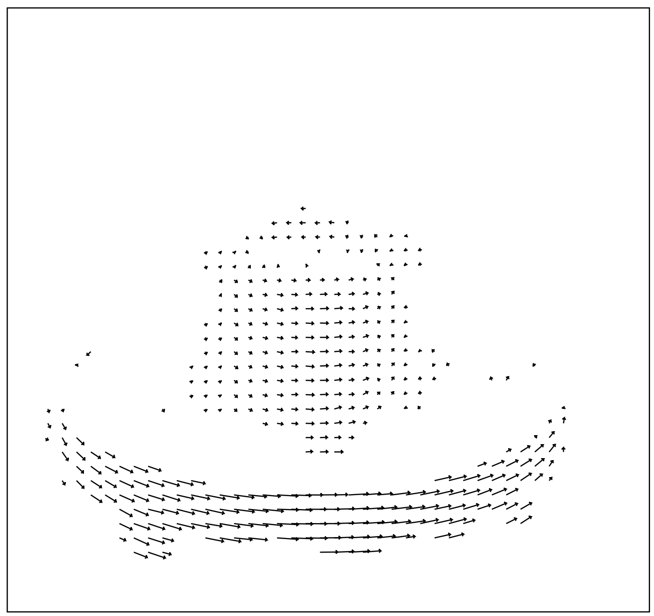
Cues from prior knowledge

Shape from...	**Assumes**
motion	rigid bodies, continuous motion
stereo	solid, contiguous, non-repeating bodies
texture	uniform texture
shading	uniform reflectance
contour	minimum curvature

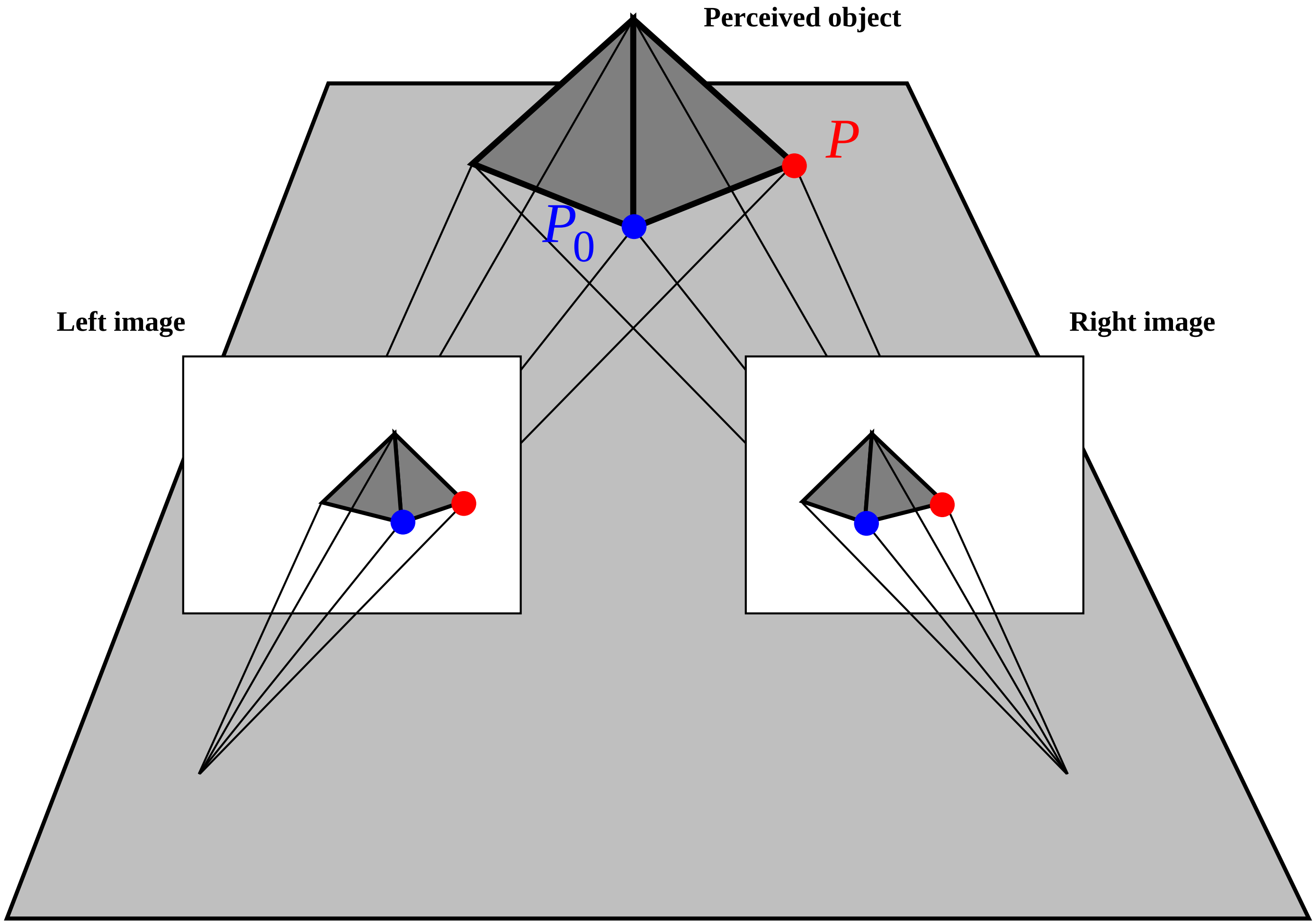
Motion



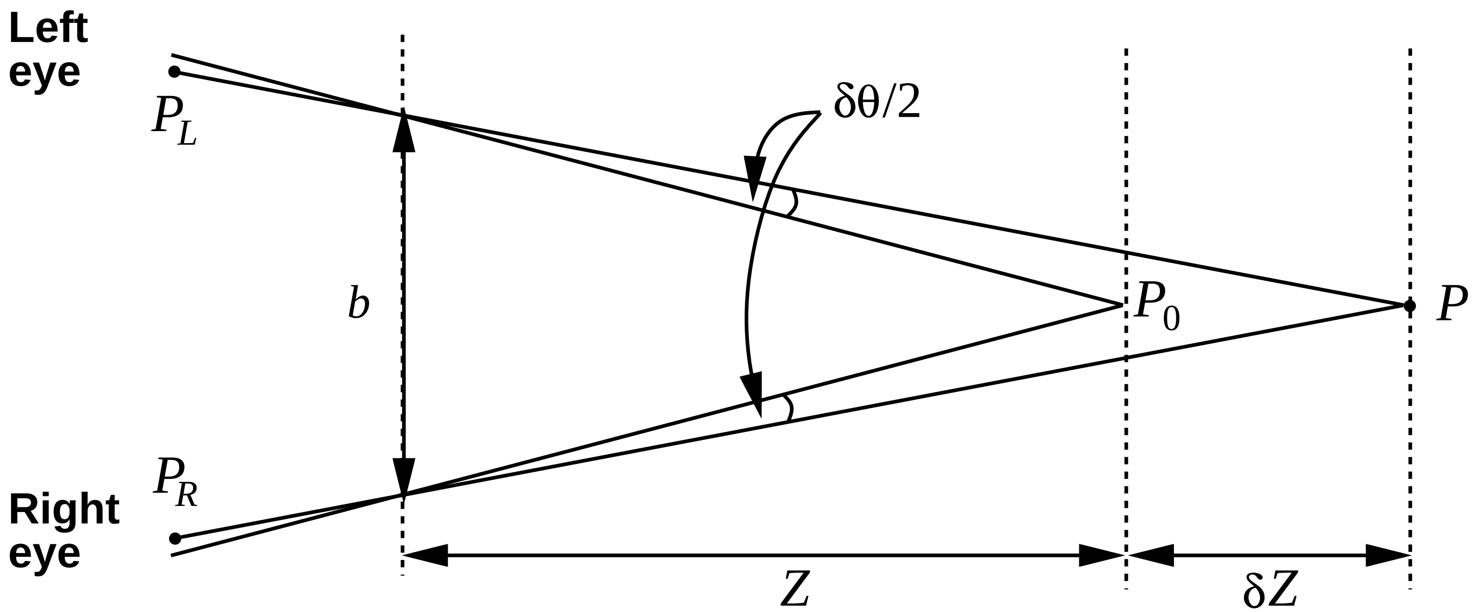




Stereo



Stereo depth resolution

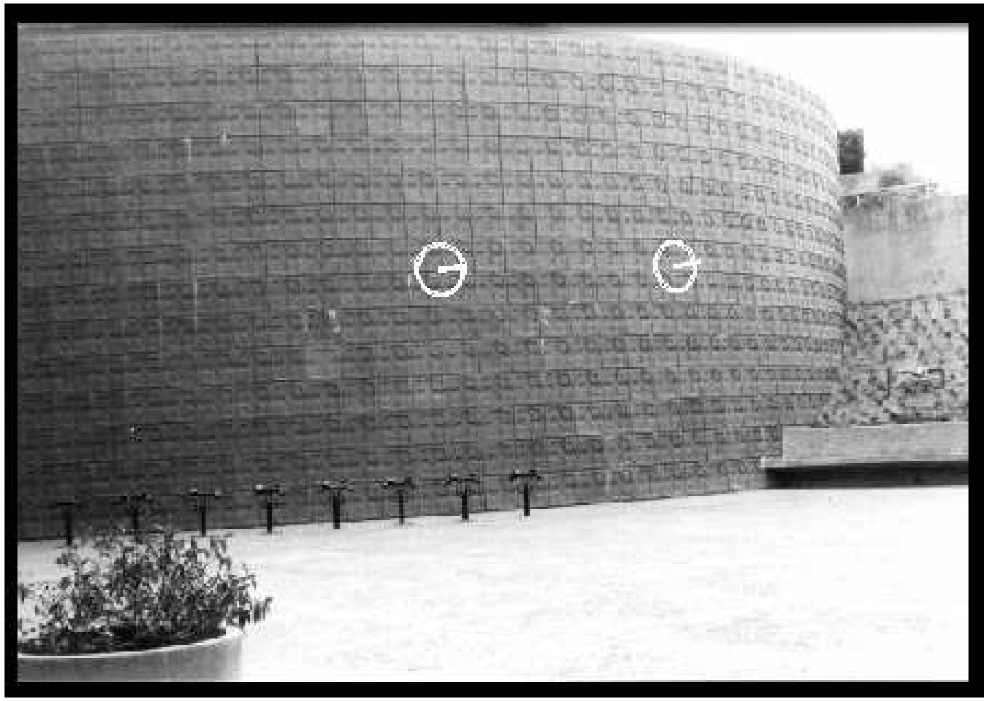


Simple geometry: $\delta Z = Z^2 \delta\theta / (-b)$
Physiology: $\delta\theta \geq 2.42 \times 10^{-5}$ radians, $b = 6$cm

$Z = 30$cm $\Rightarrow$ $\delta Z \approx 0.04$mm
$Z = 30$m $\Rightarrow$ $\delta Z \approx 40$cm

Large baseline $\Rightarrow$ better resolution!

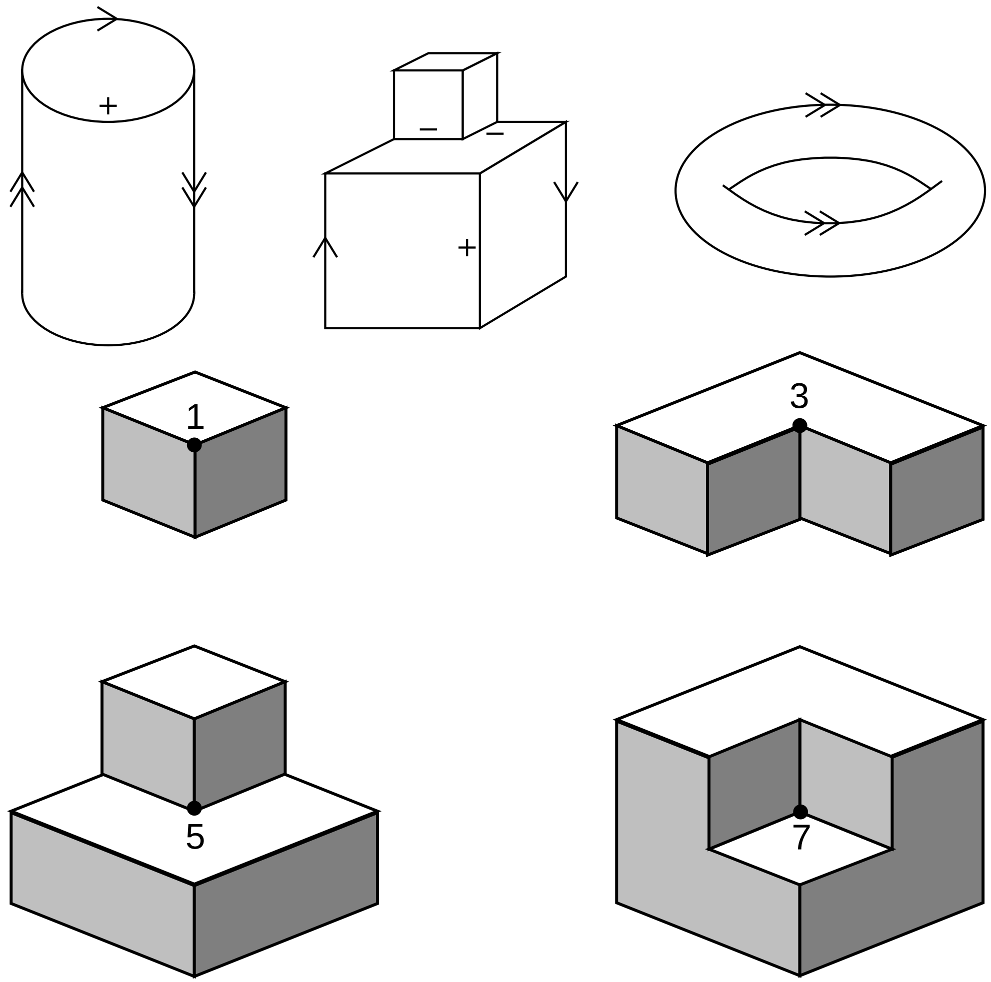
Texture



Idea: assume actual texture is uniform, compute surface shape that would produce this distortion

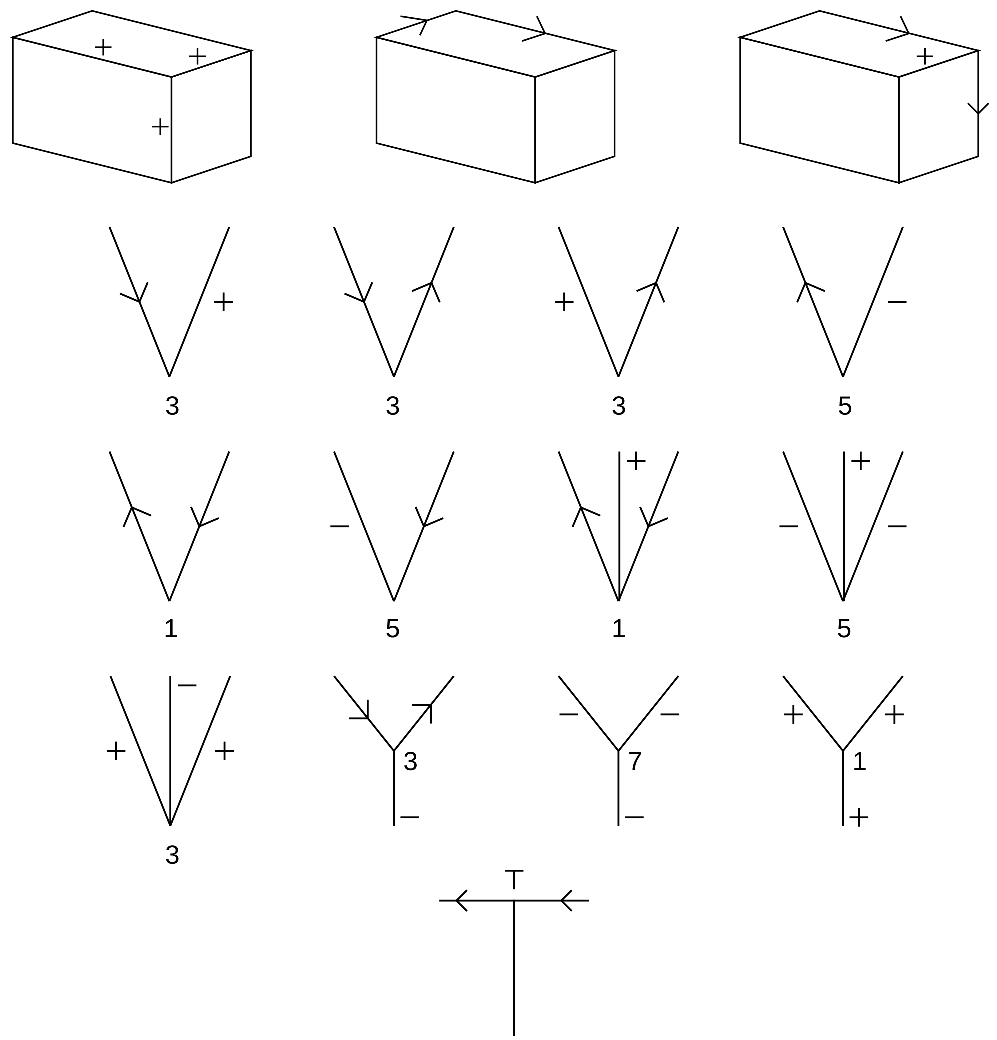
Similar idea works for shading—assume uniform reflectance, etc.—**but** interreflections give nonlocal computation of perceived intensity
⇒ hollows seem shallower than they really are

Edge and vertex types

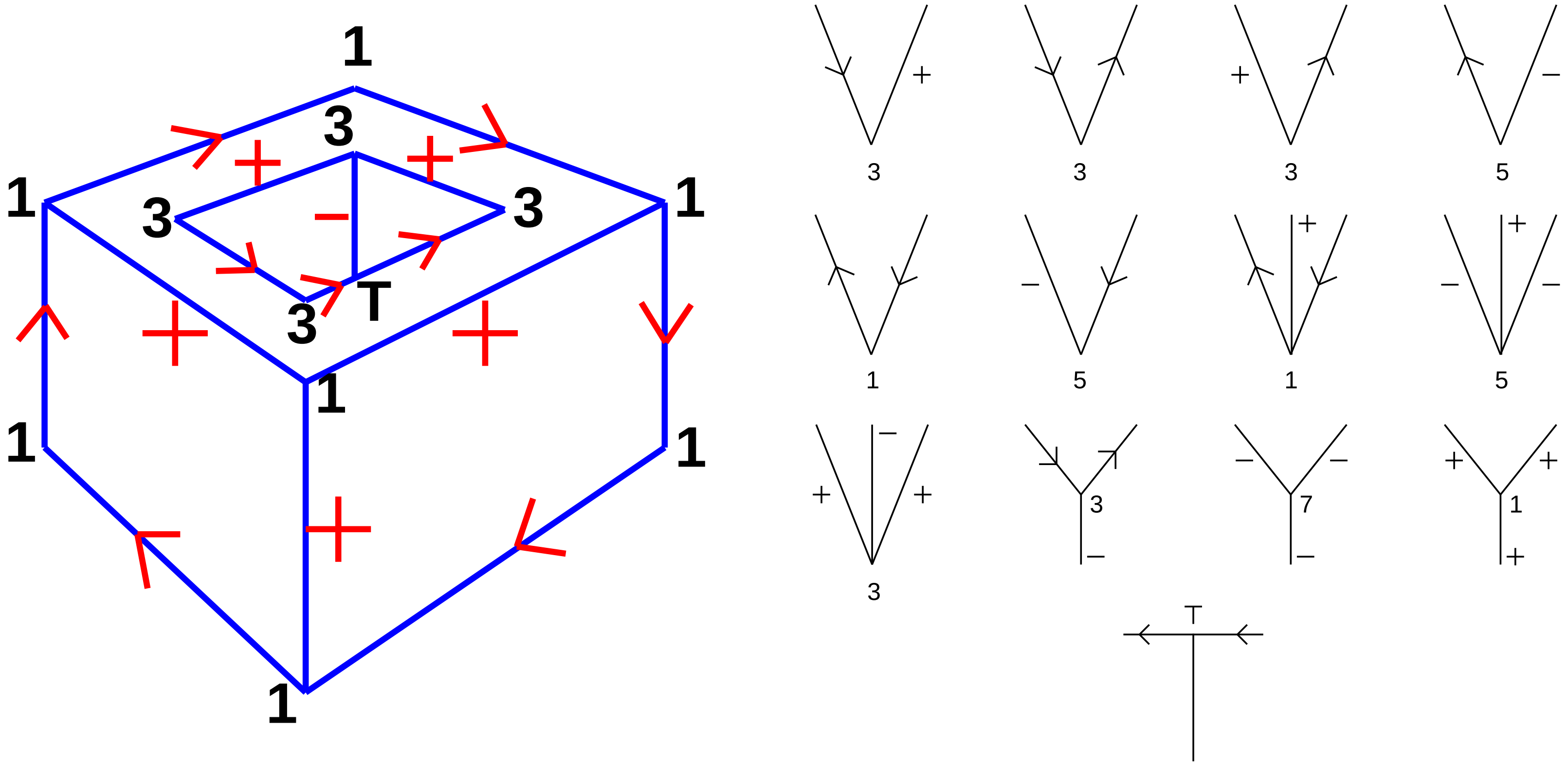


Assume world of solid polyhedral objects with trihedral vertices

Vertex/edge labels



Vertex/edge labelling example



CSP: variables = edges, constraints = possible node configurations

Object recognition

Simple idea:

– extract 3-D shapes from image
– match against "shape library"

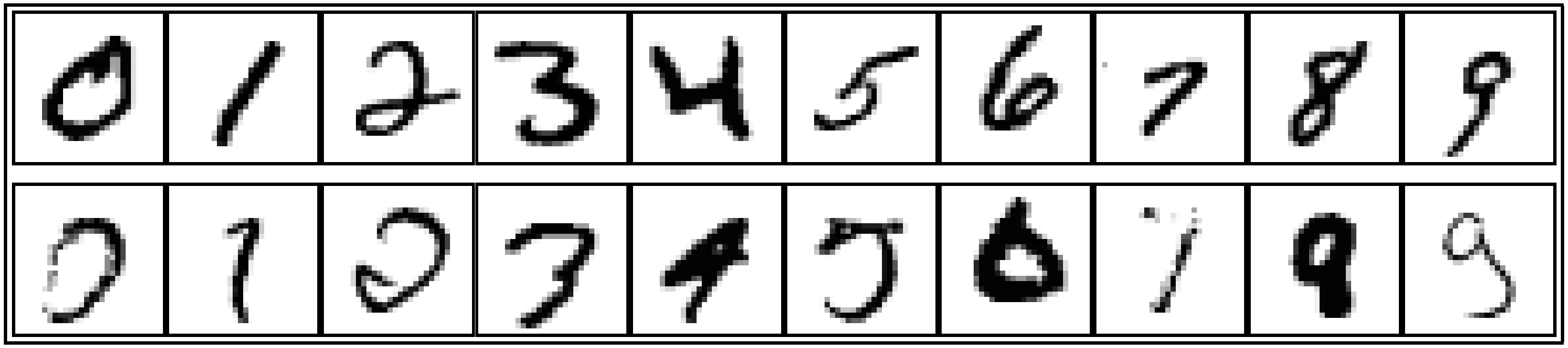
Problems:

– extracting curved surfaces from image
– representing shape of extracted object
– representing shape and variability of library object classes
– improper segmentation, occlusion
– unknown illumination, shadows, markings, noise, complexity, etc.

Approaches:

– index into library by measuring invariant properties of objects
– alignment of image feature with projected library object feature
– match image against multiple stored views (**aspects**) of library object
– machine learning methods based on image statistics

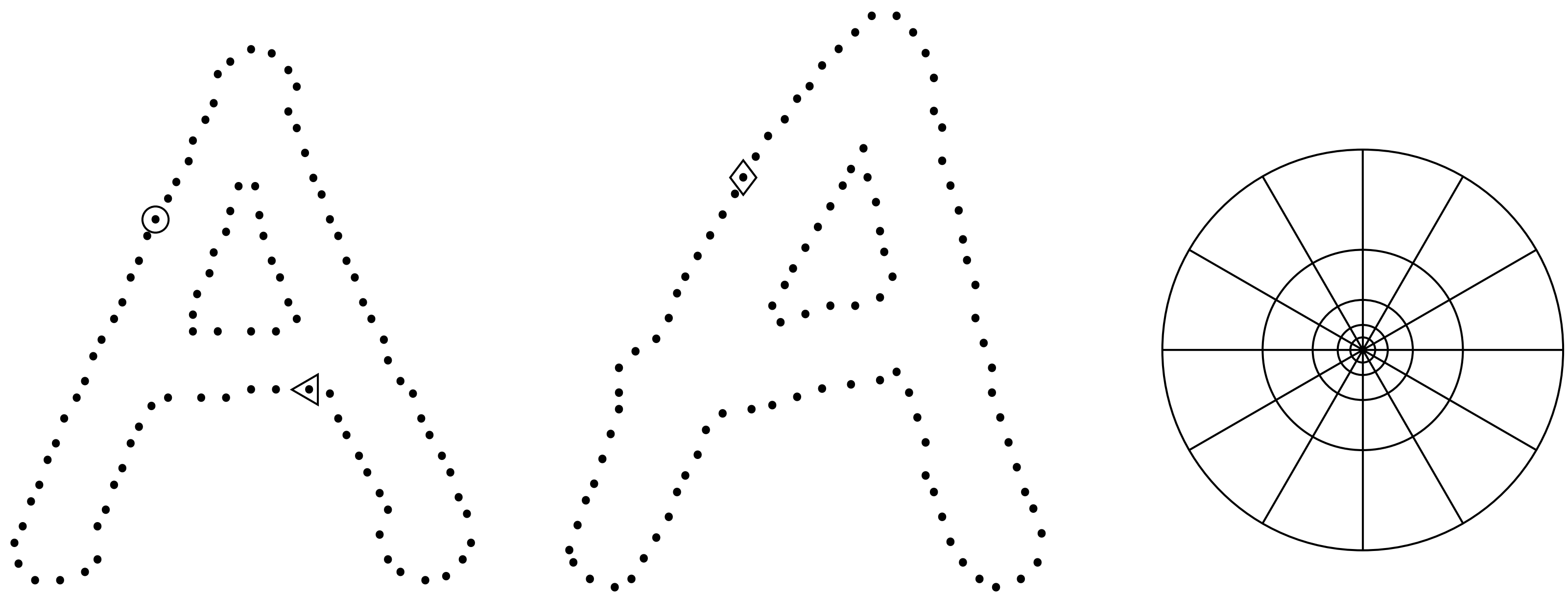
Handwritten digit recognition



3-nearest-neighbor = 2.4% error
400–300–10 unit MLP = 1.6% error
LeNet: 768–192–30–10 unit MLP = 0.9% error

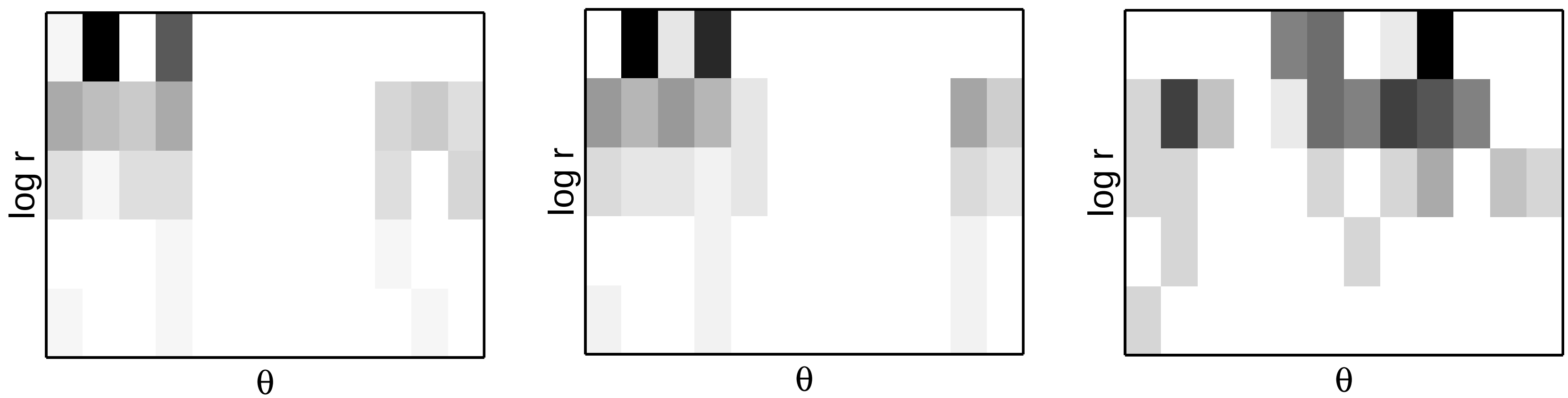
Shape-context matching

Basic idea: convert **shape** (a relational concept) into
a fixed set of **attributes** using the spatial context
of each of a fixed set of points on the surface of the shape.



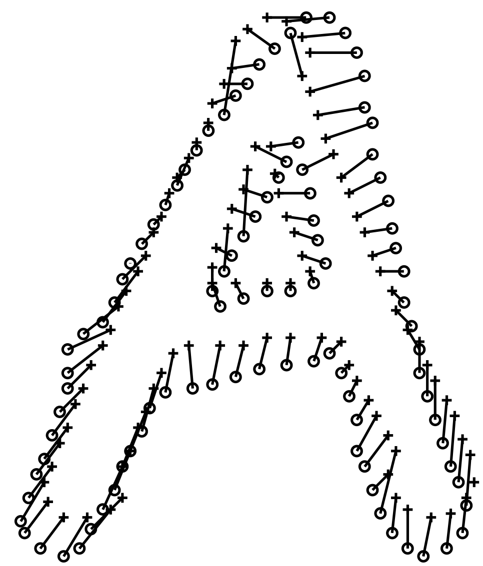
Shape-context matching contd.

Each point is described by its local context histogram
(number of points falling into each log-polar grid bin)



Shape-context matching contd.

Determine total distance between shapes by sum of distances for corresponding points under best matching



Simple nearest-neighbor learning gives 0.63% error rate on NIST digit data

Summary

Vision is hard—noise, ambiguity, complexity

Prior knowledge is essential to constrain the problem

Need to combine multiple cues: motion, contour, shading, texture, stereo

"Library" object representation: shape vs. aspects

Image/object matching: features, lines, regions, etc.