VISION

CHAPTER 24
Outline

♦ Perception generally
♦ Image formation
♦ Early vision
♦ 2D → 3D
♦ Object recognition
Perception generally

**Stimulus** (percept) $S$, **World** $W$

$$S = g(W)$$

E.g., $g = \text{“graphics.”}$ 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:

\[ P(W|S) = \alpha \frac{P(S|W)}{P(W)} \]

“graphics” “prior knowledge”

Better still: no need to recover exact scene!
Just extract information needed for
- navigation
- manipulation
- recognition/identification
Vision requires combining multiple cues
$P$ is a point in the scene, with coordinates $(X, Y, Z)$

$P'$ is its image on the image plane, with coordinates $(x, y, z)$

\[ x = \frac{-fX}{Z}, \quad y = \frac{-fY}{Z} \]

by similar triangles. Scale/distance is indeterminate!
$I(x, y, t)$ 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 $\rightarrow$ infinite-dimensional signal

Human eye has three types of color-sensitive cells; each integrates the signal $\Rightarrow$ 3-element vector intensity
Edges in image ⇐ discontinuities in scene:
1) depth
2) surface orientation
3) reflectance (surface markings)
4) illumination (shadows, etc.)
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

<table>
<thead>
<tr>
<th>Shape from...</th>
<th>Assumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>motion</td>
<td>rigid bodies, continuous motion</td>
</tr>
<tr>
<td>stereo</td>
<td>solid, contiguous, non-repeating bodies</td>
</tr>
<tr>
<td>texture</td>
<td>uniform texture</td>
</tr>
<tr>
<td>shading</td>
<td>uniform reflectance</td>
</tr>
<tr>
<td>contour</td>
<td>minimum curvature</td>
</tr>
</tbody>
</table>
Stereo

Perceived object

Left image

Right image

$P_0$

$P$
Stereo depth resolution

Simple geometry: \( \delta Z = Z^2 \delta \theta / (-b) \)

Physiology: \( \delta \theta \geq 2.42 \times 10^{-5} \) radians, \( b = 6 \text{ cm} \)

\( Z = 30 \text{ cm} \Rightarrow \delta Z \approx 0.04 \text{ mm} \)
\( Z = 30 \text{ m} \Rightarrow \delta Z \approx 40 \text{ cm} \)

Large baseline \( \Rightarrow \) better resolution!
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
Assume world of solid polyhedral objects with trihedral vertices
Vertex/edge labels

Chapter 24
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

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

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.