Segmentation and Grouping

Outline:

• Overview
• Gestalt laws
• role of recognition processes
• excursion: competition and binocular rivalry
• Segmentation as clustering
• Fitting and the Hough Transform
Credits: major sources of material, including figures and slides were:

- Peterson, M. Object recognition processes can and do operate before figure-ground organization. Current Directions in Psychological Science, 1994.
- Jitendra Malik
- and various resources on the WWW
Different Views

- Obtain a compact representation from an image/motion sequence/set of tokens
- Should support application
- Broad theory is absent at present

- Grouping (or clustering)
  - collect together tokens that “belong together”
- Fitting
  - associate a model with tokens
  - issues
    - which model?
    - which token goes to which element?
    - how many elements in the model?
General ideas

• tokens
  • whatever we need to group (pixels, points, surface elements, etc., etc.)
• top down segmentation
  • tokens belong together because they lie on the same object

• bottom up segmentation
  • tokens belong together because they are locally coherent
• These two are not mutually exclusive
Why do these tokens belong together?
Basic ideas of grouping in humans

- Figure-ground discrimination
  - grouping can be seen in terms of allocating some elements to a figure, some to ground
  - impoverished theory

- Gestalt properties
  - elements in a collection of elements can have properties that result from relationships (Muller-Lyer effect)
    - gestaltqualitat
  - A series of factors affect whether elements should be grouped together
    - Gestalt factors
Parallelism

Symmetry

Continuity

Closure
Segmentation and Recognition

Early idea: (Marr and others)
- segmentation of the scene into surfaces essentially bottom-up (2½-D sketch)
- many computer vision systems assume segmentation happens strictly before recognition

Now:
- few people still believe that general solution to bottom-up segmentation of scene is possible
- some think that segmentation processes should provide a set of segmentations among which higher processes somehow choose
Evidence for Recognition Influencing Figure-Ground Processes

After: Mary A. Peterson (1994).

Rubin vase-faces stimulus:

- you see only one shape at a time
- spontaneous switching
Reversals of Figure-Ground Organization:

- a+c: symmetry, enclosure, relative smallness of area, suggest center is the foreground
- b+d: partial symmetry, relative smallness of area, interposition, suggest center is foreground
- a+c (inverted), b+d (upright) are rotated versions of each other

Two conditions:
1. “try to see white as foreground”
2. “try to see black as foreground”

Measurement: how long do they see what as foreground before reversal?
Results: (averaged across the 2 conditions)

- Overall, subjects perceive white region longer as foreground if it’s an object in canonical orientation.
- Durations when black object is foreground get shorter
Impoverished recognition of, e.g., upside-down faces:
The First Perceived Figure-Ground Organization:

- low vs. high denotative region (left vs. right in example), matched for area and convexity

- presented for 86 ms followed by mask

Results:
- high denotative regions seen as foreground more often when upright than when inverted (76% vs. 61%)
- works for presentation times as short as 28 ms
Combination with Symmetry Effects:

- symmetry also requires presentation for at least 28 ms
- both symmetry and recognition seem to get about equal weight in influencing figure-ground organization
Object Recognition Inputs to the Organization of 3-D Displays:

- stereogram version of stimuli: disparity can suggest high or low denotative region as foreground: cooperative vs. competitive stereograms

Results:
- cooperative case: ~90% of time high-denotative region seen as foreground
- competitive case: ~50% result
- Note: for random dot stereograms high denotative region has no advantage for becoming foreground
Peterson's Model:

- Prefigural object recognition processes carve those edges that can be detected early in visual processing into parts from both sides simultaneously, therefore indicating different sets of parts along the contour's two sides. These different sets of parts are used to access in parallel the best-fitting representations of different objects in memory.

- Outputs from these activated representations in object memory serve as inputs to figure–ground computations, along with outputs from processes assessing gestalt cues such as symmetry and convexity and depth cues of both the monocular and binocular variety. Thus, the outputs from object recognition processes combine with configural cues and classic depth cues to determine scene segmentation.
**Motivation:** ability to decide between alternatives is fundamental

**Idea:** inhibitory interaction between neuronal populations representing different alternatives is plausible candidate mechanism

The most simple system:

\[
\begin{align*}
\dot{e}_1 &= \frac{1}{\tau} (-e_1 + S(K_1 - 3e_2)) \\
\dot{e}_2 &= \frac{1}{\tau} (-e_2 + S(K_2 - 3e_1))
\end{align*}
\]

Winner-take-all (WTA) network

\[
S(x) = \begin{cases} 
\frac{100x^2}{120^2 + x^2} : x \geq 0 \\
0 : x < 0
\end{cases}
\]
The Naka-Rushton function

A good fit for the steady state firing rate of neurons in several visual areas (LGN, V1, middle temporal) in response to a visual stimulus of contrast \( P \) is given by:

\[
F(P) = \begin{cases} 
\frac{r_{\text{max}} P^N}{P_{1/2}^N + P^N} : P \geq 0 \\
0 : \text{else}
\end{cases}
\]

\( P_{1/2} \), the “semi-saturation”, is the stimulus contrast (intensity) that produces half of the maximum firing rate \( r_{\text{max}} \). \( N \) determines the slope of the non-linearity at \( P_{1/2} \).
Stationary States and Stability

\[ \dot{e}_1 = \frac{1}{\tau} (-e_1 + S(K_1 - 3e_2)), \quad \dot{e}_2 = \frac{1}{\tau} (-e_2 + S(K_2 - 3e_1)) \]

The stationary states for \( K_1=K_2=120 \):
- \( e_1 = 50, e_2 = 0 \)
- \( e_2 = 50, e_1 = 0 \)
- \( e_1 = e_2 = 20 \)

Linear stability analysis:
1) for \( e_1 = 50, e_2 = 0 \):

\[
A = \begin{pmatrix}
-\frac{1}{\tau} & 0 \\
0 & -\frac{1}{\tau}
\end{pmatrix}, \text{ with } \lambda = -1/\tau \rightarrow \text{“stable node”}
\]

2) for \( e_1 = e_2 = 20 \) (\( \tau=20\text{ms} \))

\[
A = \begin{pmatrix}
-\frac{1}{\tau} & -\frac{8}{5\tau} \\
-\frac{8}{5\tau} & -\frac{1}{\tau}
\end{pmatrix}, \text{ with } \lambda_1 = -0.13, \lambda_2 = +0.03 \rightarrow \text{“unstable saddle”}
\]
Matlab Simulation

Behavior for strong identical input: $K_1=K_2=K=120$

one unit wins the competition and completely suppresses the other
Binocular Rivalry, Bistable Percepts

\[ \tau \dot{e}_1 = -e_1 + \frac{r[K_1 - \gamma e_2]^2}{(\sigma + a_1)^2 + [K_1 - \gamma e_2]^2} \]
\[ \tau \dot{e}_2 = -e_2 + \frac{r[K_2 - \gamma e_1]^2}{(\sigma + a_2)^2 + [K_2 - \gamma e_1]^2} \]
\[ \tau_A \dot{a}_1 = (-a_1 + \beta e_1) \]
\[ \tau_A \dot{a}_2 = (-a_2 + \beta e_2) \]

Idea:
extend WTA network by slow adaptation mechanism that models neural adaptation due to slow hyperpolarizing potassium current. Adaptation acts to increase semi-saturation of Naka Rushton non-linearity

ambiguous figure

binocular rivalry

\[ S(P) \text{ (spikes/sec)} \]

\[ \text{Stimulus } P \]

- 2 0 2 4 6 8 10 12 14 100

\[ \text{L} \quad \text{R} \]

Jochen Triesch, UC San Diego, http://cogsci.ucsd.edu/~triesch
Matlab Simulation

\[
\tau \dot{e}_1 = -e_1 + \frac{r[K_1 - \gamma e_2]^2_+}{(\sigma + a_1)^2 + [K_1 - \gamma e_2]^2_+}, \quad \tau \dot{a}_1 = (-a_1 + \beta e_1)
\]

\[
\tau \dot{e}_2 = -e_2 + \frac{r[K_2 - \gamma e_1]^2_+}{(\sigma + a_2)^2 + [K_2 - \gamma e_1]^2_+}, \quad \tau \dot{a}_2 = (-a_2 + \beta e_2)
\]
Discussion of Rivalry Model

Positive:
• roughly consistent with anatomy/physiology
• offers parsimonious mechanism for different perceptual switching phenomena, in a sense it “unifies” different phenomena by explaining them with the same mechanism

Limitations:
• provides only qualitative account
• real switching behaviors are not so nice and regular and simple:
  • cycles of different durations
  • temporal asymmetries
  • rivalry: competition likely takes place in hierarchical network rather than in just one stage.
• spatial dimension was ignored
• training effects
Technique: Shot Boundary Detection

- Find the shots in a sequence of video
  - shot boundaries usually result in big differences between succeeding frames
- Strategy:
  - compute interframe distances
  - declare a boundary where these are big

- Possible distances
  - frame differences
  - histogram differences
  - block comparisons
  - edge differences
- Applications:
  - representation for movies, or video sequences
    - find shot boundaries
    - obtain “most representative” frame
  - supports search
Technique: Background Subtraction

- If we know what the background looks like, it is easy to identify “interesting bits”
- Applications
  - Person in an office
  - Tracking cars on a road
  - Surveillance

- Approach:
  - Use a moving average to estimate background image
  - Subtract from current frame
  - Large absolute values are interesting pixels
    - Trick: use morphological operations to clean up pixels
a: average image. b: background subtraction with different threshold
d: background estimated with EM, e: result of Background subtraction
Segmentation as clustering

- Cluster together (pixels, tokens, etc.) that belong together
- Agglomerative clustering
  - attach closest to cluster it is closest to
  - repeat
- Divisive clustering
  - split cluster along best boundary
  - repeat

- Point-Cluster distance
  - single-link clustering (distance between clusters is shortest distance between elements)
  - complete-link clustering (distance between clusters is longest distance between elements)
  - group-average clustering
    - (distance between clusters is distance between their averages (fast!))
  - Dendrograms
    - yield a picture of output as clustering process continues
K-Means

- Choose a fixed number of clusters (K)
- Choose cluster centers and point-cluster allocations to minimize error
- Can’t do this by search, because there are too many possible allocations.

Algorithm
- Fix cluster centers; allocate points to closest cluster
- Fix allocation; compute best cluster centers
- X could be any set of features for which we can compute a distance (careful about scaling)

\[
\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of } i\text{'th cluster}} \|x_j - \mu_i\|^2 \right\}
\]
K-means clustering using intensity alone and color alone
K-means using color alone, 11 segments
K-means using color alone, 11 segments.
K-means using colour and position, 20 segments
Graph theoretic clustering

- Represent tokens using a weighted graph.
- Define affinity matrix of edge weights
- Cut up this graph to get subgraphs with strong interior links
Measuring Affinity

Intensity

$$\text{aff}(x, y) = \exp \left\{ - \left( \frac{1}{2} \sigma_i^2 \right) \left\| I(x) - I(y) \right\|^2 \right\}$$

Distance

$$\text{aff}(x, y) = \exp \left\{ - \left( \frac{1}{2} \sigma_d^2 \right) \left\| x - y \right\|^2 \right\}$$

Texture

$$\text{aff}(x, y) = \exp \left\{ - \left( \frac{1}{2} \sigma_t^2 \right) \left\| c(x) - c(y) \right\|^2 \right\}$$
Scale affects affinity
Normalized cuts

- Idea: maximize the within cluster similarity compared to the across cluster difference
- Write graph as $V$, one cluster as $A$ and the other as $B$

- Maximize

$$\frac{\text{cut}(A,B)}{\text{assoc}(A,V)} + \frac{\text{cut}(A,B)}{\text{assoc}(B,V)}$$

- i.e. construct $A$, $B$ such that their within cluster similarity is high compared to their association with the rest of the graph

- $\text{cut}(A,B) =$ sum of weights between $A$ and $B$
- $\text{assoc}(A,V) =$ sum of weights that have one end in $A$
Figure from “Image and video segmentation: the normalised cut framework”, by Shi and Malik, copyright IEEE, 1998
Figure from “Normalized cuts and image segmentation,” Shi and Malik, copyright IEEE, 2000
Data base with Human labeled segmentations is now available (Jitendra Malik)
Choose a parametric object/some objects to represent a set of tokens

Most interesting case is when criterion is not local
  - can’t tell whether a set of points lies on a line by looking only at each point and the next.

Three main questions:
  - what object represents this set of tokens best?
  - which of several objects gets which token?
  - how many objects are there?

(you could read line for object here, or circle, or ellipse or...)
Fitting and the Hough Transform

- Purports to answer all three questions
  - in practice, answer isn’t usually all that much help
- We do for lines only
- A line is the set of points \((x, y)\) such that

\[
(sin \theta)x + (cos \theta)y + d = 0
\]

- Different choices of \(\theta, d \geq 0\) give different lines
- For any \((x, y)\) there is a one parameter family of lines through this point, given by

\[
(sin \theta)x + (cos \theta)y + d = 0
\]

- Each point gets to vote for each line in the family; if there is a line that has lots of votes, that should be the line passing through the points
Mechanics of the Hough transform

- Construct an array representing $\theta, d$
- For each point, render the curve $(\theta, d)$ into this array, adding one at each cell
- Difficulties
  - how big should the cells be? (too big, and we cannot distinguish between quite different lines; too small, and noise causes lines to be missed)
- How many lines?
  - count the peaks in the Hough array
- Who belongs to which line?
  - tag the votes
- Hardly ever satisfactory in practice, because problems with noise and cell size defeat it
tokens

votes