Image segmentation

- Goal: identify groups of pixels that go together

The goals of segmentation

- Separate image into coherent “objects”
- Group together similar-looking pixels for efficiency of further processing

"superpixels"

The goals of segmentation

- Separate image into coherent “objects”
- Group together similar-looking pixels for efficiency of further processing
  “superpixels”

What is segmentation?

- Clustering image elements that “belong together”
  - Partitioning
    - Divide into regions/sequences with coherent internal properties
  - Grouping
    - Identify sets of coherent tokens in image

Segmentation

- Compact representation for image data in terms of a set of components
- Components share “common” visual properties
- Properties can be defined at different level of abstractions

Segmentation is a global process

What are the occluded numbers?

Slide credit: Fei-Fei Li
Segmentation is a global process

Occlusion is an important cue in grouping.

... but not too global

Groupings by Invisible Completions

Magritte, 1957

* Images from Steve Lehar’s Gestalt papers
Perceptual organization

“…the processes by which the bits and pieces of visual information that are available in the retinal image are structured into the larger units of perceived objects and their interrelations”

Stephen E. Palmer, Vision Science, 1999

Gestalt Psychology

- German: Gestalt - “form” or “whole”
- Berlin School, early 20th century
  - Kurt Koffka, Max Wertheimer, and Wolfgang Köhler
- Gestalt: whole or group
  - Whole is greater than sum of its parts
  - Relationships among parts can yield new properties/features
- Psychologists identified series of factors that predispose set of elements to be grouped (by human visual system)

“I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have “327”? No. I have sky, house, and trees.”

Max Wertheimer (1880-1943)
Gestalt Psychology

WOLFGANG METZGER

LAWS OF SEEING

Laws of Seeing, Wolfgang Metzger, 1936
(English translation by Lothar Spillmann, MIT Press, 2006)

Slide credit: B. Freeman and A. Torralba

Similarity

Not grouped

Proximity

Similarity

Similarity

Common Fate

Common Region

Slide credit: B. Freeman and A. Torralba

http://chicagoist.com/attachments/chicagoist_alicia/GEESE.jpg

http://www.delivery.allposters.com/WI/223/1532/PreviewComp/SuperStock_1532R-0831.jpg

Slide credit: K. Grauman
Symmetry

Common fate

Proximity

Familiarity


Image credit: Arthus-Bertrand (via F. Durand)

Slide credit: K. Grauman

http://www.capital.edu/Resources/Images/outside6_035.jpg

Slide credit: K. Grauman

http://www.capital.edu/Resources/Images/outside6_033.jpg

Slide credit: K. Grauman

Slide credit: B. Freeman and A. Torralba
**Familiarity**

Slide credit: B. Freeman and A. Torralba

**Influences of grouping**

Slide credit: B. Freeman and A. Torralba

- Grouping influences other perceptual mechanisms such as lightness perception

**Emergence**

Slide credit: S. Lazebnik

**Gestalt cues**

- Good intuition and basic principles for grouping
- Basis for many ideas in segmentation and occlusion reasoning
- Some (e.g., symmetry) are difficult to implement in practice

http://en.wikipedia.org/wiki/Gestalt_psychology

Slide credit: J. Hays
Segmentation methods

• Segment foreground from background
• Histogram-based segmentation
• Segmentation as clustering
  – K-means clustering
  – Mean-shift segmentation
• Graph-theoretic segmentation
  – Min cut
  – Normalized cuts
• Interactive segmentation

A simple segmentation 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

Two different background removal models

Background estimate
Average over frames

Foreground estimate

Foreground estimate

EM background estimate

low thresh

high thresh

EM

Images: Forsyth and Ponce, Computer Vision: A Modern Approach
Segmentation methods

- Segment foreground from background
- Histogram-based segmentation
- Segmentation as clustering
  - K-means clustering
  - Mean-shift segmentation
- Graph-theoretic segmentation
  - Min cut
  - Normalized cuts
- Interactive segmentation

These intensities define the three groups.
We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., segment the image based on the intensity feature.
  - What if the image isn’t quite so simple?

Now how to determine the three main intensities that define our groups?
We need to cluster.
- Goal: choose three “centers” as the representative intensities, and label every pixel according to which of these centers it is nearest to.

- Best cluster centers are those that minimize SSD between all points and their nearest cluster center $c_i$:
  \[
  \sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} \| p - c_i \|^2
  \]

### Segmentation methods
- Segment foreground from background
- Histogram-based segmentation
- Segmentation as clustering
  - K-means clustering
  - Mean-shift segmentation
- Graph-theoretic segmentation
  - Min cut
  - Normalized cuts
- Interactive segmentation

### Clustering
- With this objective, it is a “chicken and egg” problem:
  - If we knew the **cluster centers**, we could allocate points to groups by assigning each to its closest center.
  - If we knew the **group memberships**, we could get the centers by computing the mean per group.

### 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
- Dendrograms
  - yield a picture of output as clustering process continues
Greedy Clustering Algorithms

Algorithm 15.3: Agglomerative clustering, or clustering by merging

Make each point a separate cluster
Until the clustering is satisfactory
    Merge the two clusters with the smallest inter-cluster distance
end

Algorithm 15.4: divisive clustering, or clustering by splitting

Construct a single cluster containing all points
Until the clustering is satisfactory
    Split the cluster that yields the two components with the largest inter-cluster distance
end

Agglomerative clustering

1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters

Agglomerative clustering

1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
Agglomerative clustering

1. Say "Every point is its own cluster"
2. Find “most similar” pair of clusters
3. Merge it into a parent cluster
4. Repeat

Common similarity/distance measures

- **P-norms**
  - City Block (L1): $||x||_1 := \sum_{i=1}^{n} |x_i|$
  - Euclidean (L2): $||x||_2 := \sqrt{\sum_{i=1}^{n} x_i^2}$
  - L-infinity: $||x||_\infty := \max (|x_1|, \ldots , |x_n|)$

- **Mahalanobis**
  - Scaled Euclidean: $d(\bar{x}, \bar{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{\sigma_i^2}}$

- **Cosine distance**
  - Similarity: $\cos(\theta) = \frac{A \cdot B}{||A|| ||B||}$

Dendograms

Dendogram formed by agglomerative clustering using single-link clustering.
**Agglomerative clustering**

How to define cluster similarity?
- Average distance between points, maximum distance, minimum distance
- Distance between means or medoids

How many clusters?
- Clustering creates a dendrogram (a tree)
- Threshold based on max number of clusters or based on distance between merges

---

**Segmentation methods**

- Segment foreground from background
- Histogram-based segmentation
- Segmentation as clustering
  - K-means clustering
  - Mean-shift segmentation
- Graph-Theoretic Segmentation
  - Min cut
  - Normalized cuts

---

**K-means clustering**

- Basic idea: randomly initialize the $k$ cluster centers, and iterate between the two steps we just saw.

  1. Randomly initialize the cluster centers, $c_1, \ldots, c_k$
  2. Given cluster centers, determine points in each cluster
     - For each point $p$, find the closest $c_i$. Put $p$ into cluster $i$
  3. Given points in each cluster, solve for $c_i$
     - Set $c_i$ to be the mean of points in cluster $i$
  4. If $c_i$ have changed, repeat Step 2

Properties
- Will always converge to some solution
- Can be a “local minimum”
  - does not always find the global minimum of objective function:
    $$\sum_{i \in \text{clusters}} \sum_{p \in \text{points in cluster } i} ||p - c_i||^2$$

---

**Agglomerative clustering**

**Good**
- Simple to implement, widespread application
- Clusters have adaptive shapes
- Provides a hierarchy of clusters

**Bad**
- May have imbalanced clusters
- Still have to choose number of clusters or threshold
- Need to use an “ultrametric” to get a meaningful hierarchy
K-means

1. Ask user how many clusters they'd like. 
   \( \text{\textit{(e.g. k=5)}} \)

2. Randomly guess k cluster Center locations

3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)

4. Each Center finds the centroid of the points it owns

Slide credit: K Grauman, A. Moore
**K-means**

1. Ask user how many clusters they'd like. (e.g. $k=5$)
2. Randomly guess $k$ cluster locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns...
5. ...and jumps there
6. ...Repeat until terminated!

**K-means clustering**

- Java demo:
  
  [http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html](http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html)

---

**K-means: pros and cons**

**Pros**
- Simple, fast to compute
- Converges to local minimum of within-cluster squared error

**Cons/issues**
- Setting $k$?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters
- Assuming means can be computed

---

**An aside: Smoothing out cluster assignments**

- Assigning a cluster label per pixel may yield outliers:
  - How to ensure they are spatially smooth!
Segmentation as clustering

Depending on what we choose as the feature space, we can group pixels in different ways.

Grouping pixels based on intensity similarity

Feature space: intensity value (1-d)

Segmentation as clustering

Depending on what we choose as the feature space, we can group pixels in different ways.

Grouping pixels based on color similarity

Feature space: color value (3-d)

Segmentation as clustering

Depending on what we choose as the feature space, we can group pixels in different ways.

Grouping pixels based on intensity similarity

Clusters based on intensity similarity don’t have to be spatially coherent.
**Segmentation as clustering**

Depending on what we choose as the feature space, we can group pixels in different ways.

Grouping pixels based on intensity+position similarity

Both regions are black, but if we also include position \((x,y)\), then we could group the two into distinct segments; way to encode both similarity & proximity.
Segmentation as clustering

- Color, brightness, position alone are not enough to distinguish all regions...

Depending on what we choose as the feature space, we can group pixels in different ways.

Grouping pixels based on texture similarity

Feature space: filter bank responses (e.g., 24-d)

Segmentation with texture features

- Find “textons” by clustering vectors of filter bank outputs
- Describe texture in a window based on texton histogram

Texture representation example

- Windows with primarily horizontal edges
- Windows with primarily vertical edges
- Windows with small gradient in both directions
- Both

Statistics to summarize patterns in small windows

**Image segmentation example**

- Texture-based regions
- Color-based regions

---

**Pixel properties vs. neighborhood properties**

These look very similar in terms of their color distributions (histograms).

How would their texture distributions compare?

---

**Material classification example**

For an image of a single texture, we can classify it according to its global (image-wide) texton histogram.

---

**Material classification example**

*Nearest neighbor* classification: label the input according to the nearest known example’s label.

\[ \chi^2(h_i, h_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \]

---

Figure from Varma & Zisserman, IJCV 2005

Manik Varma
http://www.robots.ox.ac.uk/~vgg/research/texclass/wish.html

---

Slide credit: K Grauman
**Reading Assignment #5**

- Due on 12\textsuperscript{th} of December 2017

**Segmentation methods**

- Segment foreground from background
- Histogram-based segmentation
- Segmentation as clustering
  - K-means clustering
  - Mean-shift segmentation
- Graph-theoretic segmentation
  - Min cut
  - Normalized cuts
- Interactive segmentation

Next week