What is segmentation?
- Toy example
- In our eyes
- In computer’s eye
- Image segmentation

Different views of image segmentation
- As perceptual grouping
  - Gestalt rules
- As clustering — cluster pixels into groups
  - K-means, mixture of Gaussian, mean shift, …
- As labelling — assign each pixel with a label (segment id)
  - Graph cuts, random fields, …

The goals of segmentation
- Group together similar-looking pixels for efficiency of further processing
  - “Bottom-up” process
  - Unsupervised
  - “superpixels”

Inspiration from psychology
- The Gestalt school: Grouping is key to visual perception
  - The Muller-Lyer illusion

Berkeley segmentation database:
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/


http://en.wikipedia.org/wiki/Gestalt_psychology
The Gestalt school

- Elements in a collection can have properties that result from relationships
  - “The whole is greater than the sum of its parts”

subjective contours

occlusion

familiar configuration

http://en.wikipedia.org/wiki/Gestalt.psychology

Emergence

http://en.wikipedia.org/wiki/Gestalt.psychology

Gestalt factors

- These factors make intuitive sense, but are very difficult to translate into algorithms
Segmentation as clustering

- Cluster similar pixels (features) together

K-Means for segmentation

- Pros
  - Very simple method
  - Converges to a local minimum of the error function
- Cons
  - Memory-intensive
  - Need to pick K
  - Sensitive to initialization
  - Sensitive to outliers
  - Only finds “spherical” clusters

Mean shift clustering and segmentation

- An advanced and versatile technique for clustering-based segmentation

Source: K. Grauman

http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

Mean shift algorithm
- The mean shift algorithm seeks modes or local maxima of density in the feature space

Feature space (L*u*v* color values)
Mean shift clustering

- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode

Mean shift clustering/segmentation

- Find features (color, gradients, texture, etc)
- Initialize windows at individual feature points
- Perform mean shift for each window until convergence
- Merge windows that end up near the same "peak" or mode

Mean shift segmentation results

More results

http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html
Mean shift pros and cons

- **Pros**
  - Does not assume spherical clusters
  - Just a single parameter (window size)
  - Finds variable number of modes
  - Robust to outliers

- **Cons**
  - Output depends on window size
  - Computationally expensive
  - Does not scale well with dimension of feature space

Images as graphs

- Node for every pixel
- Edge between every pair of pixels (or every pair of “sufficiently close” pixels)
- Each edge is weighted by the affinity or similarity of the two nodes

Segmentation by graph partitioning

- Break Graph into Segments
  - Delete links that cross between segments
  - Easiest to break links that have low affinity
    - similar pixels should be in the same segments
    - dissimilar pixels should be in different segments

Measuring affinity

- Suppose we represent each pixel by a feature vector \( \mathbf{x} \), and define a distance function appropriate for this feature representation
- Then we can convert the distance between two feature vectors into an affinity with the help of a generalized Gaussian kernel:

\[
\exp \left( -\frac{1}{2\sigma^2} \text{dist}(\mathbf{x}_i, \mathbf{x}_j)^2 \right)
\]

Scale affects affinity

- Small \( \sigma \): group only nearby points
- Large \( \sigma \): group far-away points
Graph cut

- Set of edges whose removal makes a graph disconnected
- Cost of a cut: sum of weights of cut edges
- A graph cut gives us a segmentation
  - What is a "good" graph cut and how do we find one?

Minimum cut

- We can do segmentation by finding the **minimum cut** in a graph
  - Efficient algorithms exist for doing this

Minimum cut example

Normalized cut

- Drawback: minimum cut tends to cut off very small, isolated components
- This can be fixed by normalizing the cut by the weight of all the edges incident to the segment
- The normalized cut cost is:
  \[
  \frac{w(A, B)}{w(A, V)} + \frac{w(B, B)}{w(B, V)}
  \]

\(w(A, B)\) = sum of weights of all edges between A and B

Normalized cut example

\[y^T(D - W)y\]

where \(y\) is an indicator vector whose value should be 1 in the \(i\)th position if the \(i\)th feature point belongs to A and a negative constant otherwise

Source: S. Setz
**Normalized cut**

- Finding the exact minimum of the normalized cut cost is NP-complete, but if we relax $y$ to take on arbitrary values, then we can minimize the relaxed cost by solving the generalized eigenvalue problem $(D - W)y = \lambda Dy$
- The solution $y$ is given by the eigenvector corresponding to the second smallest eigenvalue
- Intuitively, the $i$th entry of $y$ can be viewed as a "soft" indication of the component membership of the $i$th feature
  - Can use 0 or median value of the entries as the splitting point (threshold), or find threshold that minimizes the $\text{Ncut}$ cost

**Normalized cut algorithm**

1. Represent the image as a weighted graph $G = (V, E)$, compute the weight of each edge, and summarize the information in $D$ and $W$
2. Solve $(D - W)y = \lambda Dy$ for the eigenvector with the second smallest eigenvalue
3. Use the entries of the eigenvector to bipartition the graph
4. Recursively partition the segmented parts, if necessary

**Example result**

**Challenge**

- How to segment images that are a "mosaic of textures"?

**Using texture features for segmentation**

- Convolve image with a bank of filters

**Using texture features for segmentation**

- Convolve image with a bank of filters
- Find textons by clustering vectors of filter bank outputs

---

Using texture features for segmentation

- Convolve image with a bank of filters
- Find textons by clustering vectors of filter bank outputs
- The final texture feature is a texton histogram computed over image windows at some "local scale"


Pitfall of texture features

- Possible solution: check for "intervening contours" when computing connection weights


Example results

Results: Berkeley Segmentation Engine

http://www.cs.berkeley.edu/~fowlkes/BSE/

Normalized cuts: Pro and con

- Pros
  - Generic framework, can be used with many different features and affinity formulations
- Cons
  - High storage requirement and time complexity
  - Bias towards partitioning into equal segments

Beyond traditional segmentation

- Interactive segmentation
  - Related to image editing

Graph Cuts and Efficient N-D Image Segmentation, Boykov and Funka-Lea, IJCV'04
Beyond traditional segmentation

- High-dimensional data segmentation
  - 3D volume, video data, tensor image (DTI), ...

Zhang, Javed, & Mubarak Shah, CVPR'13

Beyond traditional segmentation

- Semantic segmentation
  - A.k.a. parsing

Tu, Chen, Yuille, & Zhu, IJCV'03

Beyond traditional segmentation

- Salient region segmentation
  - A.k.a foreground segmentation, salient object detection, ...

Jiang, Ling, Yu, & Pang, CVPR'13

Figure 1. From left to right: source images, uniqueness, fuzziness, objectness, combined results and ground-truth.

Jiang, Ling, Yu, & Pang, CVPR'13

Beyond traditional segmentation

- Model-based segmentation
  - A.k.a. detection

Ling, Atkin, Zhang, Georgescu, Szeliski, & Comaniciu, CVPR'09

Ling, Atkin, Zhang, Georgescu, Szeliski, & Comaniciu, CVPR'09