Announcements

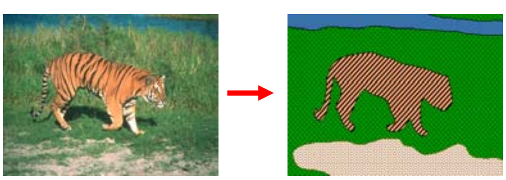
- Project 2
  - more signup slots
  - questions
- Picture taking at end of class

Image Segmentation

Today's Readings

- Forsyth chapter 14, 16

From images to objects



What Defines an Object?

- Subjective problem, but has been well-studied
- Gestalt Laws seek to formalize this
  - proximity, similarity, continuation, closure, common fate
  - see notes by Steve Joordens, U. Toronto

Image Segmentation

We will consider different methods

Already covered:

- Intelligent Scissors (contour-based)
- Hough transform (model-based)

This week:

- K-means clustering (color-based)
- EM
- Mean-shift
- Normalized Cuts (region-based)

Image histograms

How many “orange” pixels are in this image?

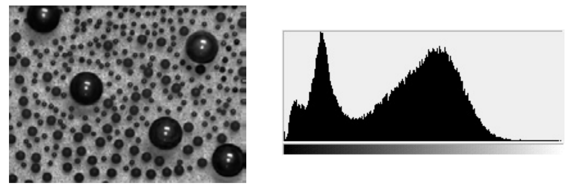
- This type of question answered by looking at the *histogram*
- A histogram counts the number of occurrences of each color
  - Given an image

$$f[x, y] \rightarrow RGB$$

  - The histogram is defined to be

$$H_f[c] = |\{(x, y) \mid f[x, y] = c\}|$$

What do histograms look like?



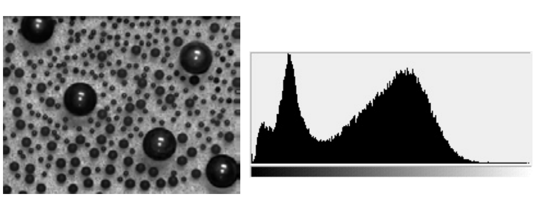
How Many Modes Are There?

- Easy to see, hard to compute

Histogram-based segmentation

Goal

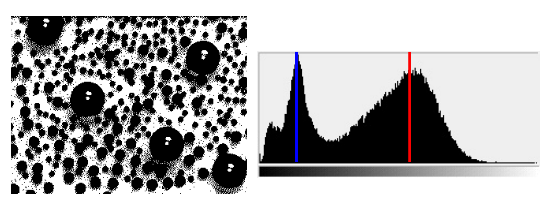
- Break the image into K regions (segments)
- Solve this by reducing the number of colors to K and mapping each pixel to the closest color
  - photoshop demo



Histogram-based segmentation

Goal

- Break the image into K regions (segments)
- Solve this by reducing the number of colors to K and mapping each pixel to the closest color
  - photoshop demo

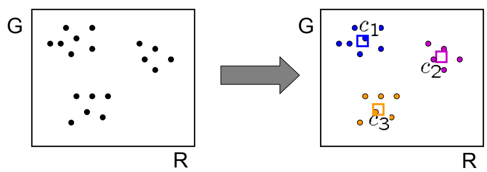


Here’s what it looks like if we use two colors

Clustering

How to choose the representative colors?

- This is a clustering problem!



Objective

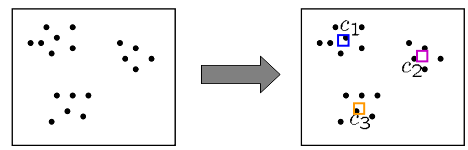
- Each point should be as close as possible to a cluster center
  - Minimize sum squared distance of each point to closest center

$$\sum_{\text{clusters } i} \sum_{\text{points p in cluster } i} ||p - c_i||^2$$

Break it down into subproblems

Suppose I tell you the cluster centers c_i

- Q: how to determine which points to associate with each c_i?
- A: for each point p, choose closest c_i



Suppose I tell you the points in each cluster

- Q: how to determine the cluster centers?
- A: choose c_i to be the mean of all points in the cluster

K-means clustering

K-means clustering algorithm

1. Randomly initialize the cluster centers, c_1, ..., 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

Java demo: http://www.cs.mcgill.ca/~bonnef/project.html

Properties

- Will always converge to *some* solution
- Can be a "local minimum"
  - does not always find the global minimum of objective function:

$$\sum_{\text{clusters } i} \sum_{\text{points p in cluster } i} ||p - c_i||^2$$

Probabilistic clustering

Basic questions

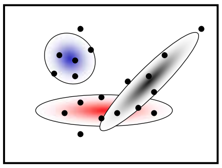
- what's the probability that a point **x** is in cluster m?
- what's the shape of each cluster?

K-means doesn't answer these questions

Basic idea

- instead of treating the data as a bunch of points, assume that they are all generated by sampling a continuous function
- This function is called a **generative model**
  - defined by a vector of parameters **θ**

Mixture of Gaussians



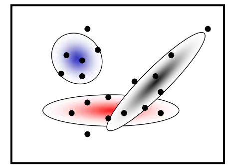
One generative model is a mixture of Gaussians (MOG)

- K Gaussian blobs with means μ_b covariance matrices V_b, dimension d
  - blob *b* defined by: $P(x|\mu_b, V_b) = \frac{1}{\sqrt{(2\pi)^d |V_b|}} e^{-\frac{1}{2}(x-\mu_b)^T V_b^{-1}(x-\mu_b)}$
- blob *b* is selected with probability α_b
- the likelihood of observing **x** is a weighted mixture of Gaussians

$$P(x|\theta) = \sum_{b=1}^{K} \alpha_b P(x|\theta_b)$$

- where $\theta = [\mu_1, \ldots, \mu_n, V_1, \ldots, V_n]$

Expectation maximization (EM)



Goal

- find blob parameters θ that maximize the likelihood function:

$$P(data|\theta) = \prod_x P(x|\theta)$$

Approach:

1. E step: given current guess of blobs, compute ownership of each point
2. M step: given ownership probabilities, update blobs to maximize likelihood function
3. repeat until convergence

EM details

E-step

- compute probability that point **x** is in blob i, given current guess of **θ**

$$P(b|x, \mu_b, V_b) = \frac{\alpha_b P(x|\mu_b, V_b)}{\sum_{i=1}^{K} \alpha_i P(x|\mu_i, V_i)}$$

M-step

- compute probability that blob b is selected

$$\alpha_b^{new} = \frac{1}{N} \sum_{i=1}^{N} P(b|x_i, \mu_b, V_b) \qquad \text{N data points}$$

- mean of blob b

$$\mu_b^{new} = \frac{\sum_{i=1}^{N} x_i P(b|x_i, \mu_b, V_b)}{\sum_{i=1}^{N} P(b|x_i, \mu_b, V_b)}$$

- covariance of blob b

$$V_b^{new} = \frac{\sum_{i=1}^{N} (x_i - \mu_b^{new})(x_i - \mu_b^{new})^T P(b|x_i, \mu_b, V_b)}{\sum_{i=1}^{N} P(b|x_i, \mu_b, V_b)}$$

EM demos

http://www.cs.ucsd.edu/users/ibayrakt/java/em/

http://www.dreier.cc/index.php?topic=downloads&sub=em

Applications of EM

Turns out this is useful for all sorts of problems

- any clustering problem
- any model estimation problem
- missing data problems
- finding outliers
- segmentation problems
  - segmentation based on color
  - segmentation based on motion
  - foreground/background separation
- ...

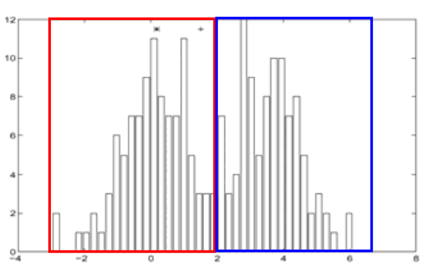
Problems with EM

Local minima

Need to know number of segments

Need to choose generative model

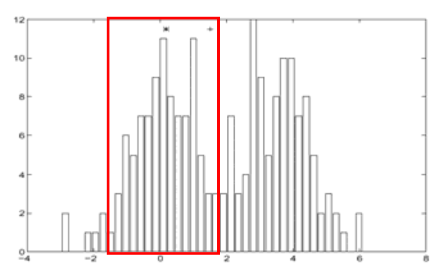
Finding Modes in a Histogram



How Many Modes Are There?

- Easy to see, hard to compute

Mean Shift [Comaniciu & Meer]



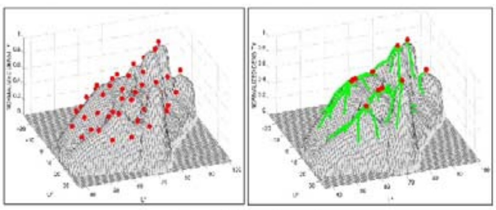
Iterative Mode Search

1. Initialize random seed, and window W
2. Calculate center of gravity (the "mean") of W: $\sum_{x \in W} xH(x)$
3. Translate the search window to the mean
4. Repeat Step 2 until convergence

Mean-Shift

Approach

- Initialize a window around each point
- See where it shifts—this determines which segment it's in
- Multiple points will shift to the same segment



Mean shift trajectories

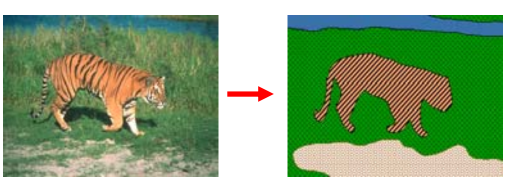
Mean-shift for image segmentation

Useful to take into account spatial information

- instead of (R, G, B), run in (R, G, B, x, y) space
- D. Comaniciu, P. Meer, Mean shift analysis and applications, *7th International Conference on Computer Vision*, Kerkyra, Greece, September 1999, 1197-1203.
  - http://www.caip.rutgers.edu/riul/research/papers/pdf/spatmsft.pdf

More Examples: http://www.caip.rutgers.edu/~comanici/segm_images.html

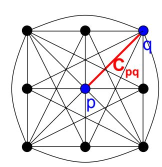
Region-based segmentation



Color histograms don't take into account spatial info

- Gestalt laws point out importance of spatial grouping
  - proximity, similarity, continuation, closure, common fate
- Suggests that *regions* are important

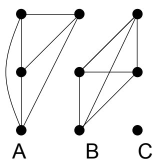
Images as graphs



Fully-connected graph

- node for every pixel
- link between *every* pair of pixels, **p**,**q**
- cost c_{pq} for each link
  - c_{pq} measures *similarity*
    - » similarity is *inversely proportional* to difference in color and position
    - » this is different than the costs for intelligent scissors

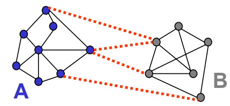
Segmentation by Graph Cuts



Break Graph into Segments

- Delete links that cross between segments
- Easiest to break links that have high cost
  - similar pixels should be in the same segments
  - dissimilar pixels should be in different segments

Cuts in a graph



Link Cut

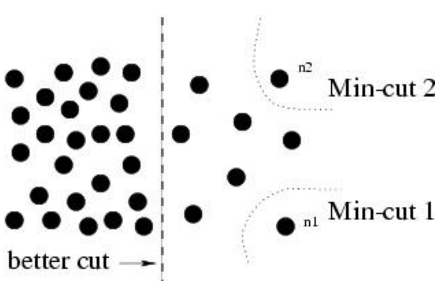
- set of links whose removal makes a graph disconnected
- cost of a cut:

$$cut(A,B) = \sum_{p \in A, q \in B} c_{p,q}$$

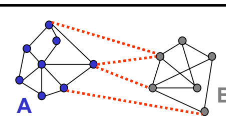
Find minimum cut

- gives you a segmentation
- fast algorithms exist for doing this

But min cut is not always the best cut...



Cuts in a graph



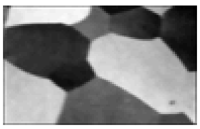
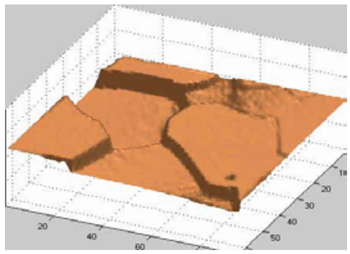
Normalized Cut

- a cut penalizes large segments
- fix by normalizing for size of segments

$$Ncut(A,B) = \frac{cut(A,B)}{volume(A)} + \frac{cut(A,B)}{volume(B)}$$

- volume(A) = sum of costs of all edges that touch A

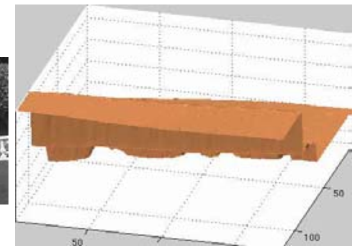
Interpretation as a Dynamical System

Treat the links as springs and shake the system

- elasticity proportional to cost
- vibration "modes" correspond to segments

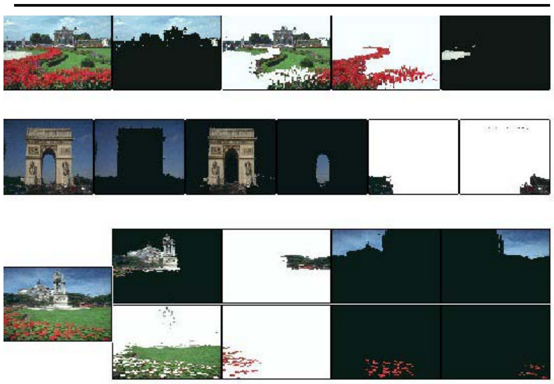
Interpretation as a Dynamical System

Treat the links as springs and shake the system

- elasticity proportional to cost
- vibration "modes" correspond to segments

Color Image Segmentation



Normalize Cut in Matrix Form

$\mathbf{W}$ is the cost matrix : $\mathbf{W}(i,j) = c_{i,j}$;

$\mathbf{D}$ is the sum of costs from node i : $\mathbf{D}(i,i) = \sum_j \mathbf{W}(i,j)$; $\mathbf{D}(i,j) = 0$

Can write normalized cut as:

$$Ncut(A,B) = \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}}, \quad \text{with } \mathbf{y}_i \in \{1,-b\}, \mathbf{y}^T\mathbf{D}1 = 0.$$

- Solution given by "generalized" eigenvalue problem:

$$(\mathbf{D}-\mathbf{W})\mathbf{y} = \lambda\mathbf{D}\mathbf{y}$$

- Solved by converting to standard eigenvalue problem:

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}\mathbf{z} = \lambda\mathbf{z}, \quad where\ \mathbf{z} = \mathbf{D}^{\frac{1}{2}}\mathbf{y}$$

- optimal solution corresponds to second smallest eigenvector
- for more details, see
  - J. Shi and J. Malik, Normalized Cuts and Image Segmentation, IEEE Conf. Computer Vision and Pattern Recognition(CVPR), 1997
  - http://www.cs.washington.edu/education/courses/455/03wi/readings/Ncut.pdf

Cleaning up the result

Problem:
- Histogram-based segmentation can produce messy regions
  - segments do not have to be connected
  - may contain holes

How can these be fixed?

photoshop demo

Dilation operator: $G = H \oplus F$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	0	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x, y]$

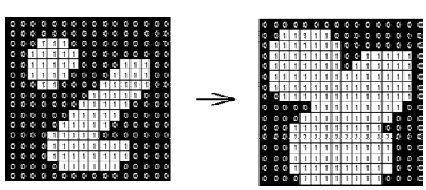
1	1	1
1	1	1
1	1	1

$H[u, v]$

Dilation: does H "overlap" F around [x,y]?
- G[x,y] = 1 if H[u,v] and F[x+u-1,y+v-1] are both 1 **somewhere**
  0 otherwise
- Written $G = H \oplus F$

Dilation operator

Demo
- http://www.cs.bris.ac.uk/~majid/mengine/morph.html



Erosion operator: $G = H \ominus F$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	0	1	1	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x, y]$

1	1	1
1	1	1
1	1	1

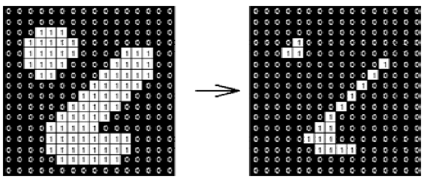
$H[u, v]$

Erosion: is H "contained in" F around [x,y]
- G[x,y] = 1 if F[x+u-1,y+v-1] is 1 **everywhere** that H[u,v] is 1
  0 otherwise
- Written $G = H \ominus F$

Erosion operator

Demo

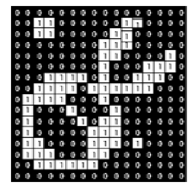
- http://www.cs.bris.ac.uk/~majid/mengine/morph.html



Nested dilations and erosions

What does this operation do?

$$G = H \ominus (H \oplus F)$$

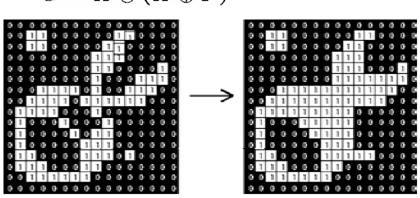


- this is called a **closing** operation

Nested dilations and erosions

What does this operation do?

$$G = H \ominus (H \oplus F)$$



- this is called a **closing** operation

Is this the same thing as the following?

$$G = H \oplus (H \ominus F)$$

Nested dilations and erosions

What does this operation do?

$$G = H \oplus (H \ominus F)$$

- this is called an **opening** operation
- http://www.dai.ed.ac.uk/HIPR2/open.htm

You can clean up binary pictures by applying combinations of dilations and erosions

Dilations, erosions, opening, and closing operations are known as **morphological operations**

- see http://www.dai.ed.ac.uk/HIPR2/morops.htm