Announcements

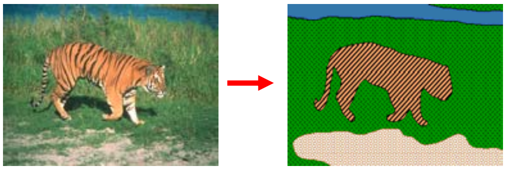
- Project 4 questions
- Evaluations

Image Segmentation

Today's Readings

- Forsyth chapter 14
- http://www.dai.ed.ac.uk/HIPR2/morops.htm
  - Dilation, erosion, opening, closing

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)

Today:

- K-means clustering (color-based)
- 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

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

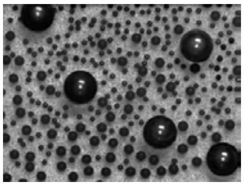
  - The histogram is defined to be

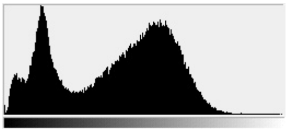
$$H_F[c] = |\{(x, y) \mid F[x, y] = c\}$$

  - What is the dimension of the histogram of an NxN RGB image?

What do histograms look like?

Photoshop demo





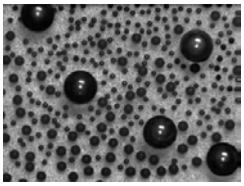
How Many Modes Are There?

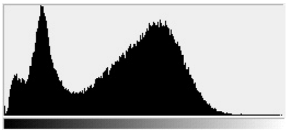
- Easy to see, hard to compute

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

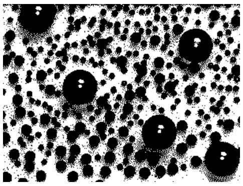


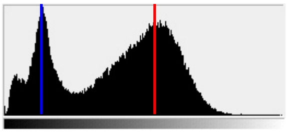


Histogram-based segmentation

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



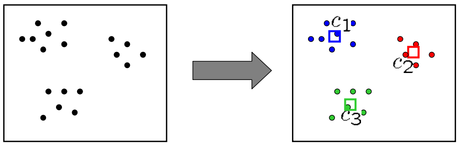


Here’s what it looks like if we use two colors

Clustering

How to choose the representative colors?

- This is a clustering problem!



Objective

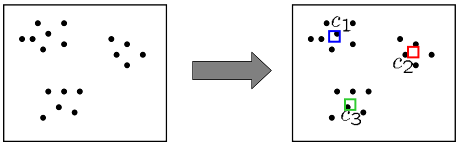
- Each point should be as close as possible to a cluster center
  - Minimize sum squared distance of each point to closest center

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

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.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/AppletKM.html

Properties

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

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

Probabilistic clustering

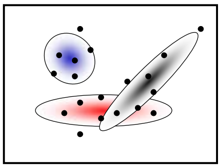
Basic questions

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

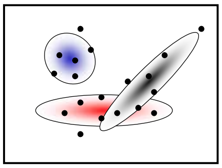
K-means doesn't answer these questions

Probabilistic clustering (basic idea)

- Treat each cluster as a Gaussian density function



Expectation Maximization (EM)



A probabilistic variant of K-means:

- E step: “soft assignment” of points to clusters
  - estimate probability that a point is in a cluster
- M step: update cluster parameters
  - mean and variance info (covariance matrix)
- maximizes the likelihood of the points given the clusters
- Forsyth Chapter 16 (optional)

EM demo

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

Applications of EM

Turns out this is useful for all sorts of problems

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

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: $G = H \oplus F$

Assume: binary image

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

$H[u, v]$

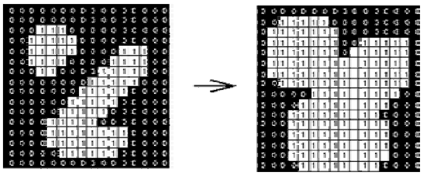
Dilation: does H “overlap” F around [x,y]?

- 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

$H[u, v]$

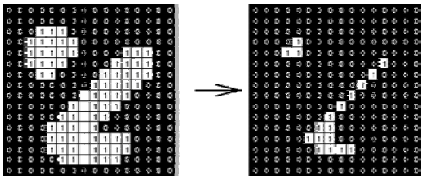
Erosion: is H “contained in” F around [x,y]

- 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

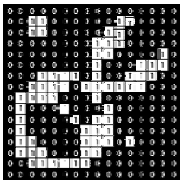
- http://www.cs.bris.ac.uk/~majid/mengine/morph.html



Nested dilations and erosions

What does this operation do?

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

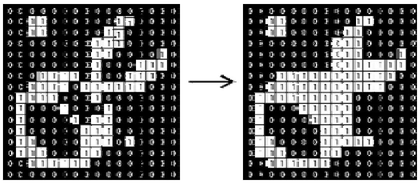


- this is called a **closing** operation

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?

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

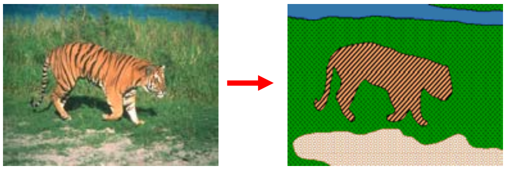
- this is called an **opening** operation
- http://www.dai.ed.ac.uk/HIPR2/open.htm

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

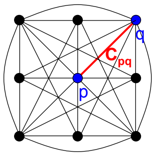
Dilations, erosions, opening, and closing operations are known as **morphological operations**

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

Graph-based segmentation?



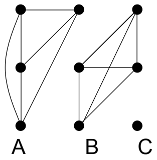
Images as graphs

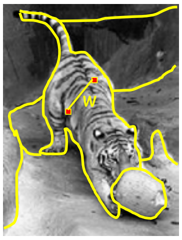


Fully-connected graph

- 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

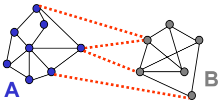




Break Graph into Segments

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

Cuts in a graph



Link Cut

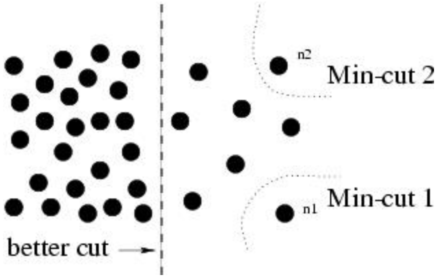
- set of links whose removal makes a graph disconnected
- cost of a cut:

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

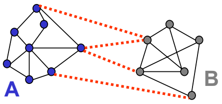
Find minimum cut

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

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



Cuts in a graph



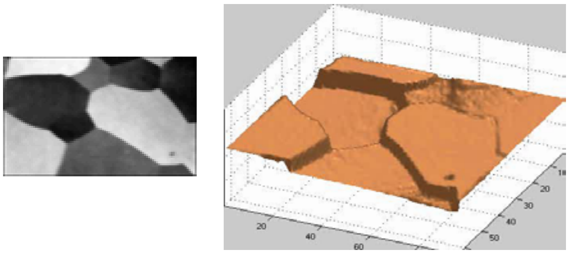
Normalized Cut

- 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

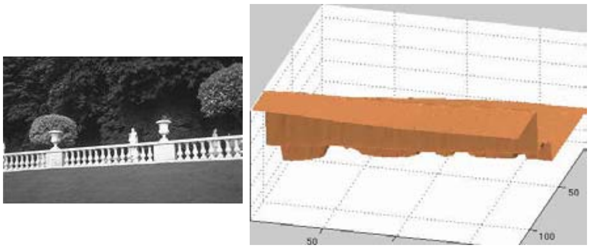
Interpretation as a Dynamical System



Treat the links as springs and shake the system

- elasticity proportional to cost
- vibration “modes” correspond to segments
  - can compute these by solving an eigenvector problem
  - Forsyth chapter 14.5

Interpretation as a Dynamical System



Treat the links as springs and shake the system

- elasticity proportional to cost
- vibration “modes” correspond to segments
  - can compute these by solving an eigenvector problem
  - Forsyth chapter 14.5

Color Image Segmentation