Announcements

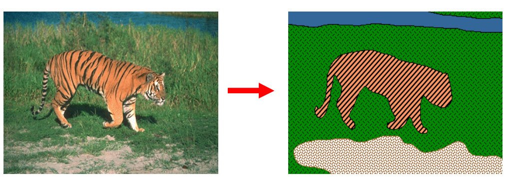
- vote for Project 3 artifacts
- Project 4 (due next Wed night)
  - Questions?
  - Late day policy: everything must be turned in by next Friday

Image Segmentation

Today's Readings

- Shapiro, pp. 279-289
- 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)

This week:

- K-means clustering (color-based)
  - Discussed in Shapiro
- Normalized Cuts (region-based)
  - Forsyth, chapter 16.5 (supplementary)

Image histograms

How many “orange” pixels are in this image?

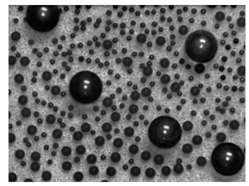
- This type of question answered by looking at the *histogram*
- A histogram counts the number of occurrences of each color
  - Given an image $F[x, y] \rightarrow RGB$
  - The histogram is defined to be

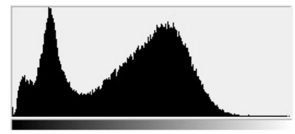
$$H_F[c] = |\{(x, y) \mid F[x, y] = c\}|$$

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

What do histograms look like?

Photoshop demo





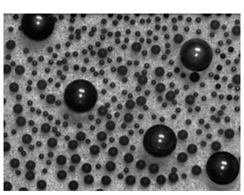
How Many Modes Are There?

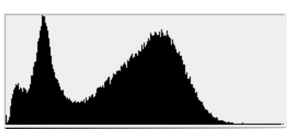
- Easy to see, hard to compute

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

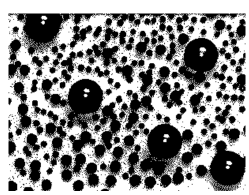


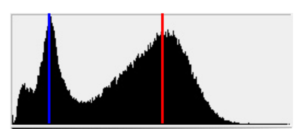


Histogram-based segmentation

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



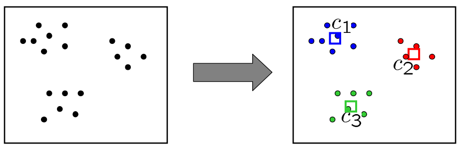


Here’s what it looks like if we use two colors

Clustering

How to choose the representative colors?

- This is a clustering problem!



Objective

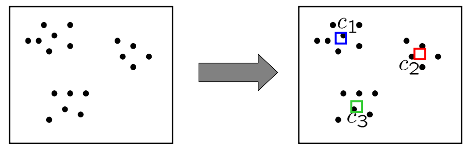
- Each point should be as close as possible to a cluster center
  - Minimize sum squared distance of each point to closest center

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

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:

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

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

$H[u, v]$

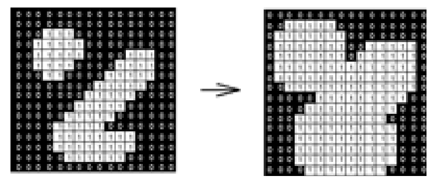
Dilation: does H "overlap" F around [x,y]?

- 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

$H[u, v]$

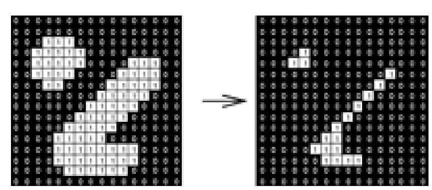
Erosion: is H "contained in" F around [x,y]

- 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

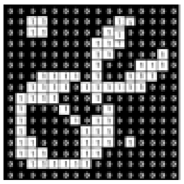
- http://www.cs.bris.ac.uk/~majid/mengine/morph.html



Nested dilations and erosions

What does this operation do?

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

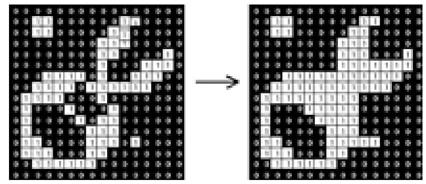


- this is called a **closing** operation

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?

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

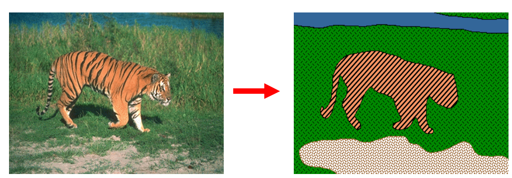
- this is called an **opening** operation
- http://www.dai.ed.ac.uk/HIPR2/open.htm

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

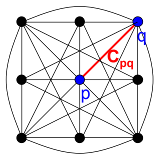
Dilations, erosions, opening, and closing operations are known as **morphological operations**

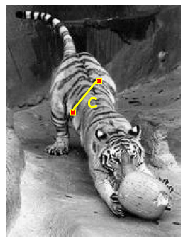
- see http://www.dai.ed.ac.uk/HIPR2/morops.htm

How about doing this automatically?



Images as graphs

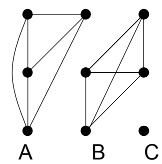


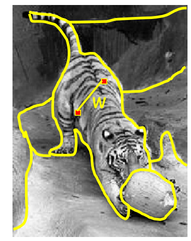


Fully-connected graph

- node for every pixel
- link between *every* pair of pixels, **p**,**q**
- cost $\mathbf{c_{pq}}$ for each link
  - $\mathbf{c_{pq}}$ 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

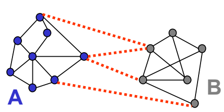




Break Graph into Segments

- 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

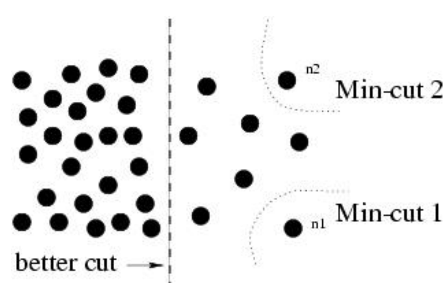
- set of links whose removal makes a graph disconnected
- cost of a cut:

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

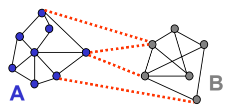
Find minimum cut

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

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



Cuts in a graph



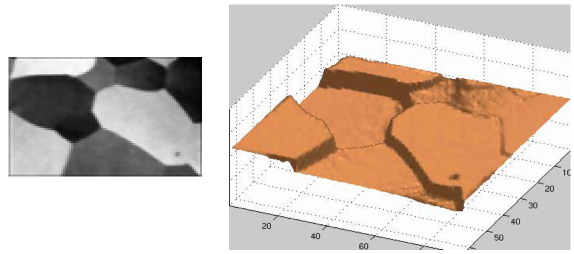
Normalized Cut

- 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

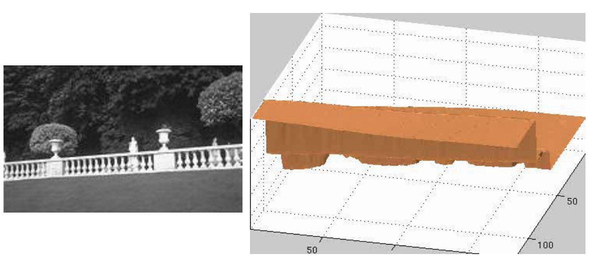
Interpretation as a Dynamical System



Treat the links as springs and shake the system

- elasticity proportional to cost
- vibration “modes” correspond to segments

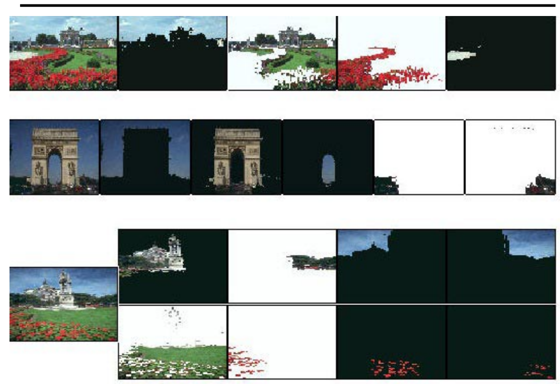
Interpretation as a Dynamical System



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:

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

- 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

Summary

Things to take away from this lecture

- Image histogram
- K-means clustering
- Morphological operations
  - dilation, erosion, closing, opening
- Normalized cuts