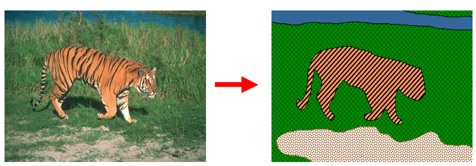
Announcements

- Questions on the project?
- Updates to project 1 page and lecture slides from 1/18
- Midterm (take home) out next Friday
  - covers material up through next Friday's lecture
  - have one week to do it
- Late policy is now online
  - 3 free late days over the quarter
  - can use on any of the projects (not midterm)
- Help session on Photoshop at the end of lecture

Segmentation (Part 2)

Today's Readings
- Shapiro, pp. 279-289 (handout)
- Watt, 10.3-10.4 (handout)
- 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 a few of these

Last Friday:
- Intelligent Scissors (contour-based)
  – E. N. Mortensen and W. A. Barrett, Intelligent Scissors for Image Composition, in ACM Computer Graphics (SIGGRAPH `95), pp. 191-198, 1995
- Normalized Cuts (region-based)
  – Discussed in Shapiro (handout), Forsyth, chapter 16.5 (supplementary)

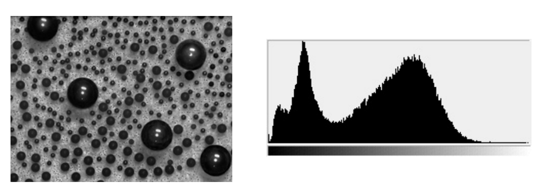
Today:
- K-means clustering (color-based)
  – Discussed in Shapiro (handout)
- Hough transform (model-based)
  – Discussed in Watt (handout)

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?

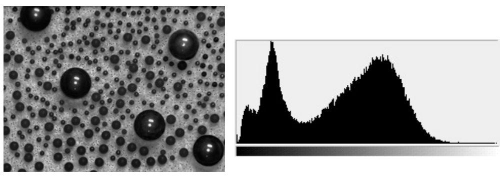
Photoshop demo



How Many Modes Are There?
- Easy to see, hard to compute

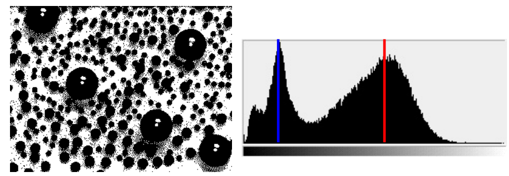
Histogram-based segmentation

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

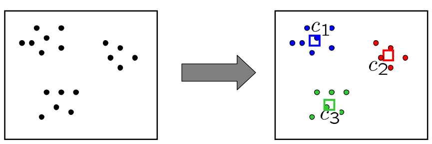
Goal
- Break the image into K regions (segments)
- Solve this by reducing the number of colors to K and mapping each pixel to the closest color
  - photoshop demo



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

Clustering

How to choose the representative colors?
- This is a clustering problem!

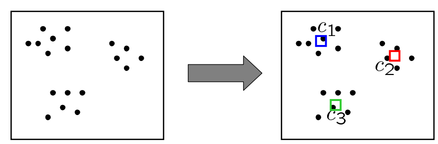


Objective
- Each point should be as close as possible to a cluster center
  - Minimize sum squared distance of each point to closest center

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

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, $c_1, \ldots, c_K$
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

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 minimum our 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

$H[u, v]$

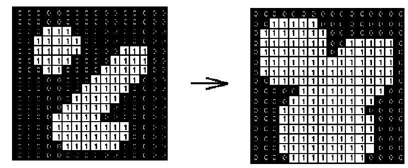
Dilation: does H "overlap" F around [x,y]?

- 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

$H[u, v]$

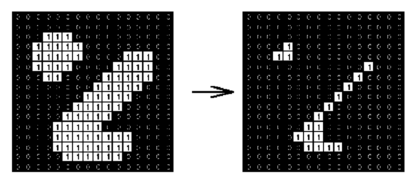
Erosion: is H "contained in" F around [x,y]

- 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

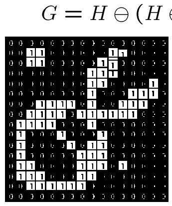
- http://www.cs.bris.ac.uk/~majid/mengine/morph.html



Nested dilations and erosions

What does this operation do?

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

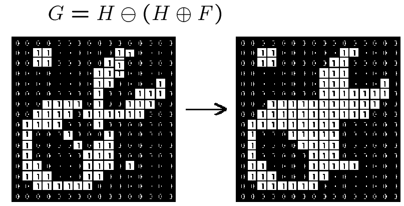


- this is called a **closing** operation

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

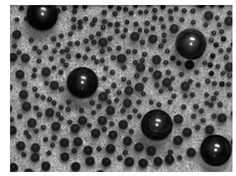
You can clean up binary pictures by applying combinations of dilations and erosions

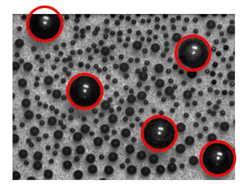
Dilations, erosions, opening, and closing operations are known as **morphological operations**

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

Model-based segmentation

Suppose we know the shapes that we're looking for?





The Hough transform

Option 1:

- Search for the object at every possible position in the image
- What is the cost of this operation?

Option 2:

- Use a voting scheme: Hough transform

Finding lines in an image

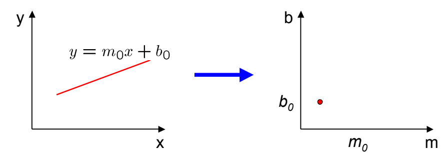


image space Hough space

Connection between image (x,y) and Hough (m,b) spaces

- A line in the image corresponds to a point in Hough space
- To go from image space to Hough space:
  - given a set of points (x,y), find all (m,b) such that y = mx + b
- What does a point in the image space map to?

Finding lines in an image

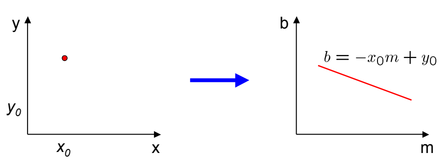


image space Hough space

Connection between image (x,y) and Hough (m,b) spaces

- A line in the image corresponds to a point in Hough space
- To go from image space to Hough space:
  - given a set of points (x,y), find all (m,b) such that y = mx + b
- What does a point (x_0, y_0) in the image space map to?
  - A: the solutions of $b = -x_0 m + y_0$
  - this is a line in Hough space

Hough transform algorithm

Typically use a different parameterization

$$d = x cos\theta + y sin\theta$$

- d is the perpendicular distance from the line to the origin
- θ is the angle this perpendicular makes with the x axis
- Why?

Basic Hough transform algorithm

1. Initialize H[d, θ]=0
2. for each edge point I[x,y] in the image

   for θ = 0 to 180

   $d = x cos\theta + y sin\theta$

   H[d, θ] += 1
3. Find the value(s) of (d, θ) where H[d, θ] is maximum
4. The detected line in the image is given by $d = x cos\theta + y sin\theta$

What's the running time (measured in # votes)?

- O(edge pixels * line directions)

Hough line demo

Extensions

Extension 1: Use the image gradient

1. same
2. for each edge point I[x,y] in the image
   compute unique (d, θ) based on image gradient at (x,y)
   H[d, θ] += 1
3. same
4. same

What's the running time measured in votes?

- O(edge pixels)

Extension 2

- give more votes for stronger edges

Extension 3

- change the sampling of (d, θ) to give more/less resolution

Extension 4

- The same procedure can be used with circles, squares, or any other shape

Hough circle demo

Summary

Things to take away from this lecture

- Graph representation of an image
- Intelligent scissors method
- Normalized cuts method
- Image histogram
- K-means clustering
- Morphological operations
  - dilation, erosion, closing, opening
- Hough transform