segmentation and clustering
segmentation vs. clustering
application: object recognition

[Perez 2002]
application: data compression

[Kaya et al. 2005]
application: image search (Blobworld)

Figure 7. Blobworld query for tiger images using two blobs. The overall weights are 1.0 for the tiger blob and 0.5 for the grass blob. For both blobs, the color weight is 1.0 and the texture weight is 0.5. (Only color is used for the background score.)

[Carson et al. 1999]
application: visual arts

[Hertzmann et al. 2006]
clustering and segmentation

• How do we define “similar regions”? 
  - Boundaries: compact, discontinuous, smooth
  - Defining similarity: color, texture, motion
  - Distance metric: minimum, mean, maximum
subjective nature of segmentation
subjective nature of segmentation
applications

foreground/background segmentation

finding skin-colored regions

finding moving objects

finding cars in a video sequence

“intelligent scissors”

semantics
applications

foreground/background segmentation

finding skin-colored regions

finding moving objects

finding cars in a video sequence

“intelligent scissors”

statistics

templates
clustering based on color

Let’s make a few concrete choices:
- arbitrary regions
- similarity based on color only
- distance between two regions equals distance between mean colors
clustering based on color

input image

four clusters
simple agglomerative clustering

- start with separate cluster for each pixel
- iterate:
  - find pair of clusters with smallest inter-cluster distance
  - merge
- stopping threshold
simple divisive clustering

- start with entire image in one cluster
- iterate:
  - find cluster with largest intra-cluster variation
  - split into two clusters that yield largest inter-cluster distance
- stopping threshold
difficulties with simple clustering

- many possibilities at each iteration
- computing distance between clusters or optimal split expensive

heuristics:
- for agglomerative clustering, approximate each cluster by average for distance computations
- for divisive clustering, use summary (histogram) of a region to compute split
**k-means clustering**

- instead of merging or splitting, start out with clusters and move them around

1. Select number of clusters $k$
2. Randomly scatter $k$ “cluster centers” in color space
3. Repeat
   3.1. Assign each data point to its closest cluster center
   3.2. Move each cluster center to the mean of the points assigned to it
$k$-means clustering

color space
$k$-means clustering
$k$-means clustering
$k$-means clustering
$k$-means clustering
$k$-means clustering
$k$-means clustering
$k$-means clustering
$k$-means clustering

original image

$k=5$

$k=11$
$k$-means clustering
other distance measures

- suppose we want compact regions?
results of clustering
other distance measures

- Problem with simple Euclidean distance: what if coordinates range from 0-1000 but colors range from 0-255?
- weighted Euclidean distance:

\[ \| x - y \|^2 = \sum_{i=1}^{D} c_i (x_i - y_i)^2 \]
Mahalanobis distance

- automatically assign weights based on statistical variation in the data:

\[ \| \bar{x} - \bar{y} \|^2 = (\bar{x} - \bar{y})' C (\bar{x} - \bar{y}) \]

where \( C \) is covariance matrix of data

- gives each dimension “equal” weight

- also accounts for correlation between different dimensions
Mahalanobis distance

\[ \| \vec{x} - \vec{y} \|^2 = (\vec{x} - \vec{y})' I (\vec{x} - \vec{y}) \]

\[ \| \vec{x} - \vec{y} \|^2 = (\vec{x} - \vec{y})' C (\vec{x} - \vec{y}) \]
graph based image segmentation
graph based image segmentation

similarity matrix
graph based image segmentation

\[ d_i = \sum_j W_{ij} \]

\[ \text{vol}(A) = \sum_{i \in A} d_i \]
graph based image segmentation

\[
cut(A, B) = \sum_{i \in A, j \in B} W_{ij}
\]

segmentation = MIN-CUT
graph based image segmentation

better cut

Min-cut 1

Min-cut 2

n1

n2
normalized cuts

\[ N\text{cut}(A, B) = \text{cut}(A, B) \left( \frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} \right) \]

NP-Hard!
normalized cuts

- approximation algorithm introduced in [Shi et al. 2000]
“fuzzy clustering”

- way of extending strict clustering (e.g., k-means) to avoid making “hard” decisions about clustering
- arises naturally from study of probability
probability and statistics in vision
probability

- objects not all the same
  - many possible shapes for people, cars, ...
  - skin has different colors
- measurements not the same due to noise
- but some measurements are more likely than others (e.g., green skin)
concrete example: detect skin

- goal: detect pixels with the color of skin
- step 1: learn distribution of skin colors from hand-labeled training data
conditional probability

- this is the probability of observing a given color given that the pixel is skin
- conditional probability: \( p(\text{color}|\text{skin}) \)
concrete example: detect skin

- Maximum Likelihood Estimation: pixel is skin if and only if $p(\text{skin}|\text{color}) > p(\text{not skin}|\text{color})$
- but, how do we compute $p(\text{skin}|\text{color})$ from $p(\text{color}|\text{skin})$?
priors

- $p(\text{skin}) = \text{prior}$
  - estimate from training data
  - tunes ‘sensitivity’ of skin detector
  - can incorporate even more information: e.g. are skin pixels more likely to be found in certain regions of the image, $p(\text{skin}|x,y)$
- with more than one class, priors encode which classes are more likely
skin detection results

[Jones and Rehg]
skin color-based face tracking
learning probability distributions

- where do probability distributions come from?
- learn them from training data
Gaussian distribution

\[ p(\vec{x}) \propto e^{-\frac{(\vec{x} - \vec{\mu})^2}{2\sigma^2}} \]
density estimation

\[ \mu = ? \]

\[ \sigma = ? \]
mixture models

\[ p(x) = \sum_{\text{classes}} \pi_{\text{class}} \ p(x|\text{class}) \]

no closed-form solution!
k-means

- iterative approach to estimating mixture model
- fixed variances!
Expectation-Maximization

- more general (iterative) approach to estimating mixture models
- applies to general “missing data” frameworks
- E-step: use known data and current estimate of model to estimate unknowns
- M-step: use current estimate of complete data to solve for optimal model
Expectation-Maximization

\[ p(x) = \sum_{\text{classes}} \pi_{\text{class}} p(x|\text{class}) \]

what are the “unknowns”?
“Probabilistic $k$-means”

- E-step: use current settings of Gaussian probabilities to estimate membership probabilities

\[
\pi_{x,j} = \frac{G_j(x)}{\sum_{j'} G_{j'}(x)}
\]

\[G_1(x; \mu_1, \sigma_1) \quad G_2(x; \mu_2, \sigma_2)\]
“Probabilistic k-means”

- **M-step**: compute parameters of model (means and variance of Gaussians) that best fit data for current memberships

\[
\begin{align*}
\mu_j &= \frac{\sum x \pi_{x,j}}{\sum \pi_{x,j}} \\
\sigma_j &= \frac{\sum (x - \mu_j)^2 \pi_{x,j}}{\sum \pi_{x,j}}
\end{align*}
\]

\[G_1(x; \mu_1, \sigma_1) \quad G_2(x; \mu_2, \sigma_2)\]
issues

outliers

hand labeling
higher-level reasoning

\[ p(\text{building}) \]
\[ p(\text{grass}) \]
\[ p(\text{car}) \]
\[ p(\text{skin}) \]
\[ p(\text{sky}) \]
\[ p(\text{indoors}) \]
\[ p(\text{RGB}|\text{building}) \]
\[ p(\text{RGB}|\text{grass}) \]
\[ p(\text{RGB}|\text{car}) \]
\[ p(\text{RGB}|\text{skin}) \]
\[ p(\text{RGB}|\text{sky}) \]
\[ p(\text{RGB}|\text{indoors}) \]
larger image neighborhoods
curse of dimensionality

How many examples for accurate density estimation?
publicly available training data

- Internet: vast collection of natural images
- converting these images into useful distributions still requires hand-labeling!
“human computation”

10s of millions of labels collected so far

ESP game

Peekaboom

common image
progress meter
message area

[von Ahn]