Computer Vision - Lecture 7
Segmentation and Grouping
19.11.2008

Bastian Leibe
RWTH Aachen
http://www.umic.rwth-aachen.de/multimedia
leibe@umic.rwth-aachen.de

Course Outline

- Image Processing Basics
- Recognition I
  - Global Representations
- Segmentation
  - Segmentation and Grouping
  - Segmentation as Energy Minimization
- Recognition II
  - Subspace representations
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking

Recap: Appearance-Based Recognition

- Basic assumption
  - Objects can be represented by a set of images ("appearances").
  - For recognition, it is sufficient to just compare the 2D appearances.
  - No 3D model is needed.

Recap: Recognition Using Histograms

- Histogram comparison

Recap: Comparison Measures

- Vector space interpretation
  - Euclidean distance
- Statistical motivation
  - Chi-square
  - Bhattacharyya
- Information-theoretic motivation
  - Kullback-Leibler divergence, Jeffreys divergence
- Histogram motivation
  - Histogram Intersection
- Ground distance
  - Earth Movers Distance (EMD)
Recap: Recognition Using Histograms

• Simple algorithm
  1. Build a set of histograms \( H = \{ h_i \} \) for each known object
  2. Compare \( h_i \) to each \( h_i \in H \)
  3. Select the object with the best matching score

“Nearest-Neighbor” strategy

Recap: Multidimensional Representations

• Combination of several descriptors
  1. Each descriptor is applied to the whole image.
  2. Corresponding pixel values are combined into one feature vector.
  3. Feature vectors are collected in multidimensional histogram.

Recap: Colored Derivatives

• Generalization: derivatives along
  - \( Y \) axis \( \rightarrow \) intensity differences
  - \( C_1 \) axis \( \rightarrow \) red-green differences
  - \( C_2 \) axis \( \rightarrow \) blue-yellow differences

• Application:
  - Brand identification in video

Recap: Histogram Backprojection

• “Where in the image are the colors we’re looking for?”
• Query: object with histogram \( M \)
• Given: image with histogram \( I \)
• Compute the “ratio histogram”:
  \[ R_i = \min \left( \frac{M}{I}, 1 \right) \]
  - \( R \) reveals how important an object color is, relative to the current image.
  - Project value back into the image (i.e. replace the image values by the values of \( R \) that they index).
  - Convolve result image with a circular mask to find the object.

Recap: Bayesian Recognition Algorithm

1. Build up histograms \( p(m_k | o) \) for each training object.
2. Sample the test image to obtain \( m_x \), \( k \in K \).
   - Only small number of local samples necessary.
3. Compute the probabilities for each training object.

\[
p(o_i | \text{image}) = \frac{\prod p(m_k | o_i) p(o_i)}{\sum_{i,k} p(m_k | o_i) p(o_i)}
\]

4. Select the object with the highest probability
   - Or reject the test image if no object accumulates sufficient probability.

Topics of This Lecture

• Segmentation and grouping
  - Gestalt principles
  - Image segmentation
• Segmentation as clustering
  - \( k \)-Means
  - Feature spaces
• Probabilistic clustering
  - Mixture of Gaussians, EM
• Model-free clustering
  - Mean-Shift clustering
• Graph theoretic segmentation
  - Normalized Cuts
Examples of Grouping in Vision

What things should be grouped?
What cues indicate groups?

Determining image regions
Grouping video frames into shots

Slide credit: Kristen Grauman
B. Leibe

Symmetry

Slide credit: Kristen Grauman
B. Leibe

Proximity

Slide credit: Kristen Grauman
B. Leibe

Muller-Lyer Illusion

Gestalt principle: grouping key to visual perception.
The Gestalt School

- Grouping is key to visual perception
- Elements in a collection can have properties that result from relationships
  - "The whole is greater than the sum of its parts"

Gestalt Theory

- Gestalt: whole or group
  - Whole is greater than sum of its parts
  - Relationships among parts can yield new properties/features
- Psychologists identified series of factors that predispose set of elements to be grouped (by human visual system)

"I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have '327'? No. I have sky, house, and trees.”

Max Wertheimer
(1880-1943)

Gestalt Factors

- These factors make intuitive sense, but are very difficult to translate into algorithms.

Continuity through Occlusion Cues

Continuity, explanation by occlusion
Continuity through Occlusion Cues

Figure-Ground Discrimination

The Ultimate Gestalt?

Image Segmentation

The Goals of Segmentation

The Goals of Segmentation
Topics of This Lecture

- Segmentation and grouping
  - Gestalt principles
  - Image Segmentation
- Segmentation as clustering
  - k-Means
  - Feature spaces
  - Probabilistic clustering
    - Mixture of Gaussians, EM
  - Model-free clustering
    - Mean Shift clustering
  - Graph theoretic segmentation
    - Normalized Cut

Image Segmentation: Toy Example

- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., segment the image based on the intensity feature.
- What if the image isn’t quite so simple?

Now how to determine the three main intensities that define our groups?
We need to cluster.

Clustering

- With this objective, it is a “chicken and egg” problem:
  - If we knew the cluster centers, we could allocate points to groups by assigning each to its closest center.
  - If we knew the group memberships, we could get the centers by computing the mean per group.
**Perceptual and Sensory Augmented Computing**

**Computer Vision WS 08/09**

**KNMeans Clustering**

- Basic idea: randomly initialize the k cluster centers, and iterate between the two steps we just saw.
  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.

- Properties
  - Will always converge to some solution.
  - Can be a "local minimum".

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

**Segmentation as Clustering**

- Java demo: [http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html](http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html)

**K-Means++**

- Can we prevent arbitrarily bad local minima?
  1. Randomly choose first center.
  2. Pick new center with prob. proportional to \( \frac{||p - c_i||^2}{\text{total error}} \).
     - Contribution of \( p \) to total error.
  3. Repeat until \( k \) centers.

- Expected error = \( O(\log k) \) * optimal

**Feature Space**

- Depending on what we choose as the feature space, we can group pixels in different ways.
- Grouping pixels based on **intensity** similarity.
- Feature space: intensity value (1D)

[Image of panda with R, G, B values listed]
Segmentation as Clustering

- Depending on what we choose as the feature space, we can group pixels in different ways.
- Grouping pixels based on texture similarity
- Feature space: filter bank responses (e.g., 24D)

Smoothing Out Cluster Assignments

- Assigning a cluster label per pixel may yield outliers:
- How can we ensure they are spatially smooth?

K-Means Clustering Results

- K-means clustering based on intensity or color is essentially vector quantization of the image attributes
  - Clusters don’t have to be spatially coherent
  - Clustering based on (r,g,b,x,y) values enforces more spatial coherence

Summary K-Means

- Pros
  - Simple, fast to compute
  - Converges to local minimum of within-cluster squared error
- Cons/Issues
  - Setting k?
  - Sensitive to initial centers
  - Sensitive to outliers
  - Detects spherical clusters only
  - Assuming means can be computed
Topics of This Lecture

- Segmentation and grouping
  - Gestalt principles
  - Image Segmentation
- Segmentation as clustering
  - Feature spaces
- Probabilistic clustering
  - Mixture of Gaussians, EM
- Model-free clustering
  - Mean-shift clustering
- Graph theoretic segmentation
  - Normalized cuts

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 \( \theta \).

Mixture of Gaussians

- One generative model is a mixture of Gaussians (MoG)
  - \( K \) Gaussian blobs with means \( \mu_b \), 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 \( P(b) \).
  - The likelihood of observing \( x \) is a weighted mixture of Gaussians
    \[
    P(x | \Theta) = \sum_{b=1}^{K} P(b | \Theta) P(x | \mu_b, V_b)
    \]

Expectation Maximization (EM)

- Goal
  - Find blob parameters \( \Theta \) that maximize the likelihood function:
    \[
    P(\text{data} | \Theta) = \prod P(x | \Theta)
    \]
- 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 \( b \), given current guess of \( \Theta \):
    \[
    P(b | x, \mu_b, V_b) = \frac{\alpha_b P(x | \mu_b, V_b)}{\sum_{b=1}^{K} \alpha_b P(x | \mu_b, V_b)}
    \]
- M-step
  - Compute probability that blob \( b \) is selected
    \[
    \alpha_b^{\text{new}} = \frac{1}{N} \sum_{i=1}^{N} P(b(i) | x(i), \mu_b, V_b)
    \]
  - Mean of blob \( b \)
    \[
    \mu_b^{\text{new}} = \frac{\sum_{i=1}^{N} x(i) P(b(i) | x(i), \mu_b, V_b)}{\sum_{i=1}^{N} P(b(i) | x(i), \mu_b, V_b)}
    \]
  - Covariance of blob \( b \)
    \[
    V_b^{\text{new}} = \frac{\sum_{i=1}^{N} (x(i) - \mu_b^{\text{new}})(x(i) - \mu_b^{\text{new}})^T P(b(i) | x(i), \mu_b, V_b)}{\sum_{i=1}^{N} P(b(i) | x(i), \mu_b, V_b)}
    \]

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
  - ...

- EM demo
Summary: Mixtures of Gaussians, EM

- **Pros**
  - Probabilistic interpretation
  - Soft assignments between data points and clusters
  - Generative model, can predict novel data points
  - Relatively compact storage

- **Cons**
  - Local minima
    - $k$-means is NP-hard even with $k=2$
  - Initialization
    - Often a good idea to start with some $k$-means iterations.
  - Need to know number of components
    - Solutions: model selection (AIC, BIC), Dirichlet process mixture
  - Need to choose generative model
  - Numerical problems are often a nuisance

Topics of This Lecture

- Segmentation and grouping
  - Gestalt principles
  - Image segmentation
- Segmentation as clustering
  - k-means
  - Feature space
- Probabilistic clustering
  - Mixture of Gaussians, EM
- Model-free clustering
  - Mean-Shift clustering
- Graph theoretic segmentation
  - Normalized Cuts

Finding Modes in a Histogram

- How many modes are there?
  - Mode = local maximum of the density of a given distribution
  - Easy to see, hard to compute

Mean-Shift Segmentation

- An advanced and versatile technique for clustering-based segmentation

Mean-Shift Algorithm

- Iterative Mode search
  1. Initialize random seed, and window $W$
  2. Calculate center of gravity (the “mean”) of $W$
  3. Shift the search window to the mean
  4. Repeat Step 2 until convergence
Mean-Shift Clustering/Segmentation

- Find features (color, gradients, texture, etc)
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge windows that end up near the same “peak” or mode

Mean-Shift Segmentation Results

http://www.cs.princeton.edu/~comanici/MSPAMI/memPamiResults.html

Real Modality Analysis

- Tessellate the space with windows
- Run the procedure in parallel

Mean-Shift Clustering

- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode

The blue data points were traversed by the windows towards the mode.

MeanNShift Clustering

- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode
**Summary Mean-Shift**

- **Pros**
  - General, application-independent tool
  - Model-free, does not assume any prior shape (spherical, elliptical, etc.) on data clusters
  - Just a single parameter (window size $h$)
    - $h$ has a physical meaning (unlike k-means)
  - Finds variable number of modes
  - Robust to outliers

- **Cons**
  - Output depends on window size
  - Window size (bandwidth) selection is not trivial
  - Computationally (relatively) expensive (~2s/image)
  - Does not scale well with dimension of feature space

---

**Images as Graphs**

- **Fully-connected graph**
  - Node (vertex) for every pixel
  - Link between every pair of pixels, $(p,q)$
  - Affinity weight $w_{pq}$ for each link (edge)
    - $w_{pq}$ measures similarity
    - Similarity is inversely proportional to difference (in color and position...)

---

**Topics of This Lecture**

- Segmentation and grouping
  - Gestalt principles
  - Image segmentation
- Segmentation as clustering
  - k-means
  - Feature space
- Probabilistic clustering
  - Mixture of Gaussians, EM
- Model-free clustering
  - Mean-Shift clustering
- Graph theoretic segmentation
  - Normalized Cuts

---

**Segmentation by Graph Cuts**

- **Break Graph into Segments**
  - Delete links that cross between segments
  - Easiest to break links that have low similarity (low weight)
  - Similar pixels should be in the same segments
  - Dissimilar pixels should be in different segments
Measuring Affinity

- Distance: \[ \text{aff}(x, y) = \exp\left( -\frac{1}{\sigma^2} \| x - y \| \right) \]
- Intensity: \[ \text{aff}(x, y) = \exp\left( -\frac{1}{\sigma^2} \| I(x) - I(y) \| \right) \]
- Color: \[ \text{aff}(x, y) = \exp\left( -\frac{1}{\sigma^2} \text{dist}(c(x), c(y))^2 \right) \]
  (some suitable color space distance)
- Texture: \[ \text{aff}(x, y) = \exp\left( -\frac{1}{\sigma^2} \| f(x) - f(y) \| \right) \]
  (vectors of filter outputs)

Scale Affects Affinity

- Small \( \sigma \): group only nearby points
- Large \( \sigma \): group far-away points

Graph Cut

- Set of edges whose removal makes a graph disconnected
- Cost of a cut
  - Sum of weights of cut edges: \( \text{cut}(A, B) = \sum_{p \in A, q \in B} w_{p,q} \)
- A graph cut gives us a segmentation
  - What is a "good" graph cut and how do we find one?

Minimum Cut

- We can do segmentation by finding the minimum cut in a graph
  - Efficient algorithms exist for doing this
- Drawback:
  - Weight of cut proportional to number of edges in the cut
  - Minimum cut tends to cut off very small, isolated components

Normalized Cut (NCut)

- A minimum cut penalizes large segments
- This can be fixed by normalizing for size of segments
- The normalized cut cost is:
  \[ N\text{cut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \]
- The exact solution cannot be computed by solving a generalized eigenvalue problem.
  \[ \text{assoc}(A, V) = \sum_{p \in A} w_{p,q} + \sum_{q \in V} w_{p,q} \]

Source: Forsyth & Ponce
**Normalized Cut (NCut)**

- A minimum cut penalizes large segments
- This can be fixed by normalizing for size of segments
- The normalized cut cost is:
  \[
  \text{NCut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} - \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}
  \]
  where \(\text{cut}(A, B) = \sum_{p \in A} w_{p,q} + \sum_{q \in B} w_{p,q}\)
  \(\text{assoc}(A, V) = \text{sum of weights of all edges in } V \text{ that touch } A\)
- The exact solution is NP-hard but an approximation can be computed by solving a generalized eigenvalue problem.

**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 a generalized eigenvector problem

**NCuts as a Generalized Eigenvector Problem**

- Definitions
  - \(W\): the affinity matrix, \(W(i, j) = w_{ij}\)
  - \(D\): the diag. matrix, \(D(i, i) = \sum W(i, j)\)
  - \(x\): a vector in \([1, -1]^N\)
- Rewriting Normalized Cut in matrix form:
  \[
  \text{NCut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} - \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}
  \]
  \[
  = \frac{(1 + x^T)(D - W)(1 + x^T)}{k^T D k} - \frac{(1 - x^T)(D - W)(1 - x^T)}{k^T D k}
  \]
  \[
  = \frac{(1 + x^T)(D - W)(1 + x^T)}{k^T D k} - \frac{(1 - x^T)(D - W)(1 - x^T)}{k^T D k}
  \]
  \[
  = \frac{\sum D(i, i)}{k^T D k}
  \]

**SomeMore Math...**

- This is hard, as \(y\) is discrete!
- Relaxation: \(\tilde{z}\) is continuous.
  \[
  \tilde{z} = (D - W)^{-1} y
  \]
  \[
  = \sum D(i, i)\]
- Optimal solution is second smallest eigenvector
- Gives continuous result—must convert into discrete values of \(y\)
NCuts: Overall Procedure

1. Construct a weighted graph $G = (V, E)$ from an image.
2. Connect each pair of pixels, and assign graph edge weights,
   $W(i, j) = \text{Prob. that } i \text{ and } j \text{ belong to the same region.}$
3. Solve $(D - W)\mathbf{v} = \lambda \mathbf{v}$ for the smallest few eigenvectors. This yields a continuous solution.
4. Threshold eigenvectors to get a discrete cut
   - This is where the approximation is made (we’re not solving NP).
5. Recursively subdivide if NCut value is below a pre-specified value.

NCuts Matlab code available at
http://www.cis.upenn.edu/~jshi/software/

Color Image Segmentation with NCuts

Results with Color & Texture

Summary: Normalized Cuts

- **Pros:**
  - Generic framework, flexible to choice of function that computes weights (“affinities”) between nodes
  - Does not require any model of the data distribution

- **Cons:**
  - Time and memory complexity can be high
  - Dense, highly connected graphs ⇒ many affinity computations
  - Solving eigenvalue problem
  - Preference for balanced partitions
  - If a region is uniform, NCuts will find the modes of vibration of the image dimensions

Segmentation: Caveats

- We’ve looked at bottom-up ways to segment an image into regions, yet finding meaningful segments is intertwined with the recognition problem.
- Often want to avoid making hard decisions too soon
- Difficult to evaluate; when is a segmentation successful?

Generic Clustering

- We have focused on ways to group pixels into image segments based on their appearance
  - Find groups; “quantize” feature space
- In general, we can use clustering techniques to find groups of similar “tokens”, provided we know how to compare the tokens.
  - E.g., segment an image into the types of motions present
  - E.g., segment a video into the types of scenes (shots) present
References and Further Reading

- Background information on segmentation by clustering and on Normalized Cuts can be found in Chapter 14 of
  - D. Forsyth, J. Ponce,
  *Computer Vision - A Modern Approach*.
  Prentice Hall, 2003

- More on the EM algorithm can be found in Chapter 16.1.2.

- Read Max Wertheimer’s classic thoughts on Gestalt
  - [http://psy.ed.asu.edu/~classics/Wertheimer/Forms/forms.htm](http://psy.ed.asu.edu/~classics/Wertheimer/Forms/forms.htm)

- Try the k-means and EM demos at
  - [http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html](http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html)

- Try the NCuts Matlab code at
  - [http://www.cis.upenn.edu/~jshi/software/](http://www.cis.upenn.edu/~jshi/software/)