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
Announcements

- No lecture on Thursday 27.11.
  - Due to an all-day faculty event
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.

⇒ Fundamental paradigm shift in the 90’s
Recap: Recognition Using Histograms

- Histogram comparison

Test image

Known objects
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
     - More exactly, for each view of each object
  2. Build a histogram $h_t$ for the test image.
  3. Compare $h_t$ to each $h_i \in H$
     - Using a suitable comparison measure
  4. Select the object with the best matching score
     - Or reject the test image if no object is similar enough.

“Nearest-Neighbor” strategy
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“: 
  \[
  R_i = \min \left( \frac{M_i}{I_i}, 1 \right)
  \]
  - \( R \) 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: Multidimensional Representations

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

\[ \begin{align*}
D_x &= 1.22 \\
D_y &= -0.39 \\
Lap &= 2.78
\end{align*} \]
Recap: Bayesian Recognition Algorithm

1. Build up histograms $p(m_k | o_n)$ for each training object.
2. Sample the test image to obtain $m_k, k \in K$.
   - Only small number of local samples necessary.
3. Compute the probabilities for each training object.
   \[
   \begin{align*}
   & m_i \quad p(o_n | m_i) \\
   & m_j \quad p(o_n | m_j) \\
   & \vdots \\
   \end{align*}
   \]
   \[
   p(o_n | Image) = \frac{\prod_k p(m_k | o_n) p(o_n)}{\sum_i \prod_k p(m_k | o_i) p(o_i)}
   \]
4. Select the object with the highest probability
   - Or reject the test image if no object accumulates sufficient probability.
Recap: Colored Derivatives

• Generalization: derivatives along
  - Y axis → intensity differences
  - C₁ axis → red-green differences
  - C₂ axis → blue-yellow differences

• Application:
  - Brand identification in video

B. Leibe

[Hall & Crowley, 2000]
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

B. Leibe
Examples of Grouping in Vision

Determining image regions

Grouping video frames into shots

What things should be grouped?

What cues indicate groups?

Object-level grouping

Figure-ground

Slide credit: Kristen Grauman
Similarity

Slide credit: Kristen Grauman
Symmetry

Slide credit: Kristen Grauman
Common Fate

Image credit: Arthus-Bertrand (via F. Durand)

Slide credit: Kristen Grauman

B. Leibe
Proximity

Slide credit: Kristen Grauman
http://www.capital.edu/Resources/Images/outside6_035.jpg

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”

Illusory/subjective contours

Occlusion

Familiar configuration

http://en.wikipedia.org/wiki/Gestalt_psychology

Slide credit: Svetlana Lazebnik

Image source: Steve Lehar
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)

http://psy.ed.asu.edu/~classics/Wertheimer/Forms/forms.htm

B. Leibe
Gestalt Factors

- Not grouped
- Proximity
- Similarity
- Similarity
- Common Fate
- Common Region

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

B. Leibe
Continuity through Occlusion Cues
Continuity through Occlusion Cues

Continuity, explanation by occlusion

B. Leibe
Continuity through Occlusion Cues

Image source: Forsyth & Ponce
Continuity through Occlusion Cues
Figure-Ground Discrimination
The Ultimate Gestalt?
Image Segmentation

• Goal: identify groups of pixels that go together
The Goals of Segmentation

- Separate image into coherent “objects”
The Goals of Segmentation

• Separate image into coherent “objects”

• Group together similar-looking pixels for efficiency of further processing

“superpixels”


Slide credit: Svetlana Lazebnik
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
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?
Input image

Pixel count

Intensity

Input image

Pixel count

Intensity

Slide credit: Kristen Grauman
• Now how to determine the three main intensities that define our groups?
• We need to cluster.
Goal: choose three “centers” as the representative intensities, and label every pixel according to which of these centers it is nearest to.

Best cluster centers are those that minimize SSD between all points and their nearest cluster center $c_i$:

$$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} ||p - c_i||^2$$
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.
K-Means 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”
    - Does not always find the global minimum of objective function:
      $$\sum_{clusters \ i} \sum_{points \ p \ in \ cluster \ i} \|p - c_i\|^2$$

Slide credit: Steve Seitz
Segmentation as Clustering

```matlab
img_as_col = double(im(:));
cluster_membs = kmeans(img_as_col, K);

labelim = zeros(size(im));
for i=1:k
    inds = find(cluster_membs==i);
    meanval = mean(img_as_column(inds));
    labelim(inds) = meanval;
end
```

Slide credit: Kristen Grauman

B. Leibe
K-Means Clustering

- Java demo:
  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 $\frac{1}{\|p - c_i\|^2}$ (Contribution of $p$ to total error)
3. Repeat until $k$ centers.

- Expected error = $O(\log k) \times$ optimal

Arthur & Vassilvitskii 2007
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)
Feature Space

- Depending on what we choose as the feature space, we can group pixels in different ways.

- Grouping pixels based on color similarity

- Feature space: color value (3D)

Slide credit: Kristen Grauman  
B. Leibe
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?

Slide credit: Kristen Grauman
Segmentation as Clustering

- Depending on what we choose as the feature space, we can group pixels in different ways.

- Grouping pixels based on intensity + position similarity

⇒ Way to encode both similarity and proximity.

Slide credit: Kristen Grauman

B. Leibe
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
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

Slide credit: Kristen Grauman
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
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 $\alpha_b$
  - The likelihood of observing $x$ is a weighted mixture of Gaussians
    $$P(x|\theta) = \sum_{b=1}^{K} \alpha_b P(x|\theta_b), \quad \theta = [\mu_1, \ldots, \mu_n, V_1, \ldots, V_n]$$

Slide credit: Steve Seitz
Expectation Maximization (EM)

- **Goal**
  - Find blob parameters $\theta$ that maximize the likelihood function:
  \[
P(data|\theta) = \prod_x P(x|\theta)
  \]

- **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 $\theta$
    \[
    P(b|x, \mu_b, V_b) = \frac{\alpha_b P(x|\mu_b, V_b)}{\sum_{i=1}^{K} \alpha_i P(x|\mu_i, V_i)}
    \]

- **M-step**
  - Compute probability that blob $b$ is selected
    \[
    \alpha_b^{new} = \frac{1}{N} \sum_{i=1}^{N} P(b|x_i, \mu_b, V_b)
    \quad \text{(N data points)}
    \]
  - Mean of blob $b$
    \[
    \mu_b^{new} = \frac{\sum_{i=1}^{N} x_i P(b|x_i, \mu_b, V_b)}{\sum_{i=1}^{N} P(b|x_i, \mu_b, V_b)}
    \]
  - Covariance of blob $b$
    \[
    V_b^{new} = \frac{\sum_{i=1}^{N} (x_i - \mu_b^{new})(x_i - \mu_b^{new})^T P(b|x_i, \mu_b, V_b)}{\sum_{i=1}^{N} P(b|x_i, \mu_b, V_b)}
    \]
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
Segmentation with EM

Original image

EM segmentation results

k=2  k=3  k=4  k=5

Image source: Serge Belongie
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 spaces
- 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?
  - \textit{Mode} = local maximum of the density of a given distribution
  - Easy to see, hard to compute

Slide credit: Steve Seitz
Mean-Shift Segmentation

- An advanced and versatile technique for clustering-based segmentation

http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html


Slide credit: Svetlana Lazebnik

B. Leibe
Mean-Shift Algorithm

- **Iterative Mode Search**
  1. Initialize random seed, and window $W$
  2. Calculate center of gravity (the “mean”) of $W$: $\sum_{x \in W} x H(x)$
  3. Shift the search window to the mean
  4. Repeat Step 2 until convergence
Mean-Shift

Region of interest

Center of mass

Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean-Shift

Region of interest
Center of mass
Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean-Shift

Region of interest

Center of mass

Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean-Shift

Region of interest
Center of mass

Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean-Shift

Region of interest
Center of mass
Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean-Shift

Region of interest
Center of mass
Real Modality Analysis

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

Slide by Y. Ukrainitz & B. Sarel
The blue data points were traversed by the windows towards the mode.
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
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

Slide credit: Svetlana Lazebnik
Mean-Shift Segmentation Results

http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

Slide credit: Svetlana Lazebnik
More Results
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
  - Does not scale well with dimension of feature space
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

B. Leibe
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...)

Slide credit: Steve Seitz
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**  \( af(x, y) = \exp \left\{ -\frac{1}{2\sigma_d^2} \| x - y \|^2 \right\} \)

- **Intensity**  \( af(x, y) = \exp \left\{ -\frac{1}{2\sigma_d^2} \| I(x) - I(y) \|^2 \right\} \)

- **Color**  \( af(x, y) = \exp \left\{ -\frac{1}{2\sigma_d^2} \text{dist}(c(x), c(y))^2 \right\} \)
  (some suitable color space distance)

- **Texture**  \( af(x, y) = \exp \left\{ -\frac{1}{2\sigma_d^2} \| f(x) - f(y) \|^2 \right\} \)
  (vectors of filter outputs)

Source: Forsyth & Ponce
Scale Affects Affinity

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

Slide credit: Svetlana Lazebnik

Image Source: Forsyth & Ponce
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?

Slide credit: Steve Seitz
Here, the cut is nicely defined by the block-diagonal structure of the affinity matrix.

⇒ How can this be generalized?
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

Cuts with lesser weight than the ideal cut

Ideal Cut

Slide credit: Khurram Hassan-Shafique
Normalized Cut (NCut)

- A minimum cut penalizes large segments
- This can be fixed by normalizing for size of segments
- The normalized cut cost is:

\[
Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}
\]

\[
assoc(A, V) = \sum \text{sum}
\]

\[
= cut(A, B) \left[ \frac{1}{\sum p \in A w_{p,q}} + \frac{1}{\sum q \in B w_{p,q}} \right]
\]

- The exact solution is NP-hard but an approximation can be computed by solving a generalized eigenvalue problem.

Normalized Cut (NCut)

- A minimum cut penalizes large segments
- This can be fixed by normalizing for size of segments
- The normalized cut cost is:

\[
N\text{cut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}
\]

\[
\text{assoc}(A, V) = \text{sum of weights of all edges in } V \text{ that touch } A
\]

- The exact solution is NP-hard but an approximation can be computed by solving a \textit{generalized eigenvalue} problem.


Slide credit: Svetlana Lazebnik
B. Leibe
• 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

\[ W : \text{the affinity matrix, } W(i, j) = w_{i,j} ; \]
\[ D : \text{the diag. matrix, } D(i,i) = \sum_j W(i, j) ; \]
\[ x : \text{a vector in } \{1, -1\}^N, x(i) = 1 \iff i \in A. \]

• Rewriting Normalized Cut in matrix form:

\[ NCut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \]
\[ = \frac{(1+x)^T (D-W)(1+x)}{k1^T D1} + \frac{(1-x)^T (D-W)(1-x)}{(1-k)1^T D1} ; \quad k = \frac{\sum_{x_i > 0} D(i,i)}{\sum_i D(i,i)} \]
\[ = \ldots \]
SomeMore Math...

We see again this is an unbalanced measure, which reflects how tightly on average, nodes within the group are connected to each other.

A rather important property of this definition of association and disassociation of a partition is that they are not actually related:

\[
N_{\text{ass}}(A, B) = \frac{\text{Cput}(A, B) + \text{Cput}(B, A)}{\text{Cput}(A, V) + \text{Cput}(B, V)}
\]

\[
N_{\text{dis}}(A, B) = \frac{\text{Ccut}(A, B) - \text{Ccut}(A, V) - \text{Ccut}(B, V)}{\text{Ccut}(A, V) + \text{Ccut}(B, V)}
\]

\[
N_{\text{ass}}(A, B) = 2 - N_{\text{dis}}(A, B)
\]

Hence, the two partition criteria that we seek in our grouping algorithm, maximizing the association between the groups and minimizing the association of the group to itself in fact identical, and can be satisfied simultaneously. In our algorithm, we will use this normalization of the partition criterion.

Having defined the graph partition criterion that we want to optimize, we will show how such an optimal partition can be computed efficiently.

2.1 Computing the optimal partition

Given a partition of nodes of a graph, \( G \), into two sets \( A \) and \( \bar{A} \), let \( \sum \) be an \( N \times N \) dimensional indicator matrix, \( \sum_{i=1}^{N} (x \in A \land i \neq k) \). Let \( N_{\text{ass}}(A, B) = \sum \delta_{i \in \bar{A}} \delta_{j \in A} \). Let \( D \) to be the total connection from nodes \( i \) to all other nodes. With the definitions above and the connection matrix, we can write \( N_{\text{ass}}(A, B) \) as:

\[
N_{\text{ass}}(A, B) = \frac{\sum_{i \in \bar{A}} \sum_{j \in A} \delta_{i \in \bar{A}} \delta_{j \in A}}{\sum_{i \in \bar{A}} \sum_{j \in \bar{A}} \delta_{i \in \bar{A}} \delta_{j \in \bar{A}}}
\]

Let \( D \) be an \( N \times N \) diagonal matrix with \( d_{i,i} \) on the diagonal, \( W \) be an \( N \times N \) symmetrical matrix with \( W_{i,j} = \frac{1}{d_{i,j}} \), \( \sum W_{i,j} = 1 \), and \( 1 \) be an \( N \times 1 \) vector of ones. Then, the cut of \( A \) and \( \bar{A} \) can be expressed as:

\[
\text{Ccut}(A, B) = \sum_{i \in A} \sum_{j \in \bar{A}} W_{i,j}
\]

\[
\text{Ccut}(B, A) = \sum_{j \in A} \sum_{i \in \bar{A}} W_{i,j}
\]

\[
\text{Ccut}(A, V) = \sum_{i \in A} \sum_{j \in V} W_{i,j}
\]

\[
\text{Ccut}(B, V) = \sum_{j \in A} \sum_{i \in V} W_{i,j}
\]

Let \( c_{i} = \sqrt{d_{i,i}} \), \( (\cdot)^{\frac{1}{2}} \), \( \frac{1}{d_{i,j}} \), \( \sum_{i \in A} \delta_{i} \), and \( M = \sum_{i \in A} \delta_{i} \), we can then further expand the above equation as:

\[
\frac{-\gamma c_{i}^{2} + 2(1-2\gamma)\max(c_{i}) + 2(1-2\gamma)\bar{c}_{i} + 2\gamma}{M}
\]

dropping the last constant term, which in this case equals 0, we get:

\[
\frac{-\gamma c_{i}^{2} + 2(1-2\gamma)\max(c_{i}) + 2(1-2\gamma)\bar{c}_{i} + 2\gamma}{M}
\]

\[
\frac{2\gamma c_{i}^{2} + 2(1-2\gamma)\max(c_{i}) + 2(1-2\gamma)\bar{c}_{i} + 2\gamma}{M}
\]

\[
\frac{2\gamma c_{i}^{2} + 2(1-2\gamma)\max(c_{i}) + 2(1-2\gamma)\bar{c}_{i} + 2\gamma}{M}
\]

\[
\frac{2\gamma c_{i}^{2} + 2(1-2\gamma)\max(c_{i}) + 2(1-2\gamma)\bar{c}_{i} + 2\gamma}{M}
\]

\[
\frac{2\gamma c_{i}^{2} + 2(1-2\gamma)\max(c_{i}) + 2(1-2\gamma)\bar{c}_{i} + 2\gamma}{M}
\]

\[
\frac{2\gamma c_{i}^{2} + 2(1-2\gamma)\max(c_{i}) + 2(1-2\gamma)\bar{c}_{i} + 2\gamma}{M}
\]

\[
\frac{2\gamma c_{i}^{2} + 2(1-2\gamma)\max(c_{i}) + 2(1-2\gamma)\bar{c}_{i} + 2\gamma}{M}
\]

\[
\frac{2\gamma c_{i}^{2} + 2(1-2\gamma)\max(c_{i}) + 2(1-2\gamma)\bar{c}_{i} + 2\gamma}{M}
\]

\[
\frac{2\gamma c_{i}^{2} + 2(1-2\gamma)\max(c_{i}) + 2(1-2\gamma)\bar{c}_{i} + 2\gamma}{M}
\]
NCuts as a Generalized Eigenvalue Problem

- After simplification, we get

\[ \text{NCut}(A, B) = \frac{y^T (D - W) y}{y^T D y}, \quad \text{with } y_i \in \{1, -b\}, \ y^T D 1 = 0. \]

- This is a Rayleigh Quotient
  - Solution given by the “generalized” eigenvalue problem
    \[ (D - W) y = \lambda D y \]
  - Solved by converting to standard eigenvalue problem
    \[ D^{-\frac{1}{2}} (D - W) D^{-\frac{1}{2}} z = \lambda z, \quad \text{where } z = D^{\frac{1}{2}} y \]

- Subtleties
  - Optimal solution is second smallest eigenvector
  - Gives continuous result—must convert into discrete values of \( y \)
NCuts Example

Smallest eigenvectors

NCuts segments

B. Leibe

Image source: Shi & Malik
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 \( (D - W)y = \lambda Dy \) 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/

Slide credit: Jitendra Malik
Color Image Segmentation with NCuts

Slide credit: Steve Seitz

Image Source: Shi & Malik
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

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

• Try the NCuts Matlab code at 
  http://www.cis.upenn.edu/~jshi/software/