Image Analysis

Segmentation by Clustering

Christophoros Nikou

cnikou@cs.uoi.gr

Images taken from:
Computer Vision course by Svetlana Lazebnik, University of North Carolina at Chapel Hill.
Computer Vision course by Kristen Grauman, University of Texas at Austin.

1

Image Segmentation

• Obtain a compact representation of the image to be used for further processing.
• Group together similar pixels
• Image intensity is not sufficient to perform semantic segmentation
  – Object recognition
    • Decompose objects to simple tokens (line segments, spots, corners)
  – Finding buildings in images
    • Fit polygons and determine surface orientations.
  – Video summarization
    • Shot detection

2

Image Segmentation (cont.)

– Bottom-up or top-down process?
– Supervised or unsupervised?

Berkeley segmentation database:
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench

C. Nikou – École Doctorale, Strasbourg, 6-10 June 2011

3

Image Segmentation (cont.)

Goal: separate an image into "coherent" regions.
• Major approaches
  – Basic techniques (thresholding, region growing, morphological watersheds).
  – Clustering.
  – Model fitting.
  – Probabilistic methods.
Basic techniques also follow a clustering principle. We will examine more advanced clustering methods later.

4

Segmentation by Clustering

• Grouping and the Gestalt school.
• K-means
• Mean shift
• Spectral methods
  – Average association
  – Average cut
  – Normalized cut
• Textons
• Segmentation as a first step to image understanding.

C. Nikou – École Doctorale, Strasbourg, 6-10 June 2011

5

Segmentation and Grouping

• Image analysis and computer vision are inference problems
  – We have measurements and a model.
  – We would like to know what caused the measurement.
• Obtain a compact representation from an image/motion sequence/set of tokens.
• Grouping (or clustering)
  – collect together tokens (pixels, points surface elements...) that "belong together".
• Fitting
  – associate a model with tokens.
  – issues
    • which model?
    • which token goes to which element?
    • how many elements in the model?
• Should support application.
• Broad theory is absent at present.

C. Nikou – École Doctorale, Strasbourg, 6-10 June 2011

6
Perceptual Organization

• Why do these tokens belong together?
• The human visual system performs surprisingly well.
• How could we make a computer system “see” as a human?

Basic ideas of grouping in humans

The Gestalt (“shape”) school: Grouping is the key to human visual perception.

• Figure-ground discrimination
  - grouping can be seen in terms of allocating some elements to a figure, some to ground.
  - impoverished theory.

• Gestalt properties
  - elements in a collection of elements can have properties that result from relationships (Muller-Lyer effect)
  - gestaltqualitat
  - A series of factors affect whether elements should be grouped together.

The Gestalt School

Psychologists identified series of factors that predispose sets of elements to be grouped in the 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 house, sky and trees.”

Max Wertheimer (1880-1943).
Untersuchungen zur Lehre von der Gestalt.

The Gestalt School (cont.)

• Figure-ground discrimination
  - White circle on a black background?
  - A black rectangle with a hole on it?

The Gestalt School (cont.)

• Subjective contours
  - Scattered tokens?
  - Occlusion?

The Gestalt School (cont.)

• The Muller-Lyer illusion:
  - You can’t look at this figure and ignore the arrowheads.
• Elements in a collection can have properties that result from relationships.
  – “The whole is greater than the sum of its parts”

Intuitive factors leading to grouping are very difficult to translate into algorithms.

Gestalt factors in natural images

Symmetry

Common fate
Gestalt factors in natural images (cont.)

Proximity

What can you see?

Gestalt factors (cont.)

The visual system is helped by the evidence that the tokens are separated for a reason: occlusion.

What can you see?

Gestalt factors (cont.)

Continuity through occlusion (is it a cube?)

Elevator buttons at Computer Science building, U.C. at Berkeley.
Gestalt factors (cont.)

Even psychologists argued which factors are more dominant than the others.
Today, Gestalt factors are important but they should be considered as consequences of a more general grouping mechanism and not as the mechanism itself.
Which is this mechanism?
- We cannot answer yet.

Application: Background Subtraction

Simple segmentation algorithms work well when we know what to look for.
If we know what the background looks like, it is easy to identify “interesting bits”.
Applications
- Person in an office
- Tracking cars on a road
- Surveillance

Approach:
- Use a weighted moving average (over time) to estimate background image.
- Distant frames are assigned smaller weights (e.g., the weather changes smoothly from rain to sunshine).
- This is a filter smoothing a function of time.
- Subtract from current frame.
- Large absolute values are interesting pixels.
- The method is powerful at coarse scales.

Every 5th frame of the sequence is shown.
The child moves from one side to the other.

Background estimation averaging frames of size 80x60. The child spent more time on the right side.

Pixels of a frame whose difference from the average exceed a threshold.
In both thresholds, there are excess pixels and missing pixels.

Background estimation by a more sophisticated method (EM).

Pixels of a frame whose difference from the average exceed a threshold.
There are also excess pixels and missing pixels.
The high frequency texture of the sofa pattern was mistaken for the child. This is because small movements can cause misregistration of spatial content carrying high frequencies. The same results at a higher resolution (160x120).

Video sequences are composed of shots:
- Shorter subsequences showing largely the same object.
- It is very helpful to represent a video as a collection of shots
  - Each shot is represented by a key frame.
  - Video browsing retrieval.
- Approach: find frames that are significantly different from the previous frame.
  - Take into account that object and background may move.
  - Simple difference will not do the job.

Other approaches:
- Histogram based methods
  - Compute color histograms and take the difference.
  - Insensitive to motion and camera jitters.
- Block comparison
  - Avoids difficulties of color histograms.
    - A red object disappearing in the bottom is equivalent to a red object appearing in the top.
- Edge differencing
  - Compare edge maps (corresponding edges between frames).
- These are ad hoc methods but usually sufficient for standard problems.

Most image segmentation algorithms are based on clustering.
- Agglomerative clustering
  - Each data item is regarded as a cluster.
  - Clusters are recursively merged.
- Divisive clustering
  - The entire data set is regarded as a cluster.
  - Clusters are recursively split.

What is a good inter-cluster distance?
- The distance between the closest elements in the clusters (single-link clustering)
  - Tends to form extended clusters.
- The maximum distance between an element of the first and one of the second cluster (complete-link clustering)
  - Tends to provide rounded clusters.
- The average distance between elements in the clusters (group average clustering)
  - Tends to form rounded clusters also.

Major issue:
- How many clusters are there?
  - Difficult to answer if there is no model for the process that generated the data.
  - The hierarchy may be displayed in the form of a dendrogram.
    - A representation of cluster structure displaying cluster distances.
    - We may determine the number of clusters from the dendrogram.
Segmentation by Clustering (cont.)

• A dendrogram obtained by agglomerative clustering.
• Selecting a particular value of distance then a horizontal line at that distance splits the dendrogram into clusters.
• It gives some insight into how good are the clusters.

Common distances in image analysis involve color, texture and difference in position (to provide blobby segments).

Problem for using agglomerative or divisive clustering: there are a lot of pixels in the image.
  – Too big dendrograms.
  – Impractical to look for the best split (merge) of clusters.
  • Divisive methods are modified by using a summary of a cluster (histogram).
  • Agglomerative methods also use coordinates for inter-cluster distance computation, and merging is performed on neighboring clusters.

K-Means for Segmentation

• Choose a fixed number of clusters.
• Choose cluster centers and point-cluster allocations to minimize error
• We can’t do this by search, because there are too many possible allocations.
• It minimizes the dispersion of the data from the centers.

\[
E(\text{clusters, data}) = \sum_{i \in \text{clusters}} \left( \sum_{j \in i \text{ cluster}} (x_j - c_i)^T (x_j - c_i) \right)
\]

K-Means for Segmentation (cont.)

• Cluster similar pixels (features) together

Image Intensity-based clusters Color-based clusters

• Segments are not necessarily connected using only color.
• Absence of texture features is obvious (red cabbage).
K-Means for Segmentation (cont.)

- Cluster similar pixels (features) together

\[
\begin{align*}
R=0 & \quad G=200 \\
B=2 & \quad X=50 \\
Y=20 & \\
\vdots & \\
R=15 & \quad G=189 \\
B=2 & \quad X=20 \\
Y=400 & \\
R=3 & \quad G=12 \\
B=2 & \quad X=100 \\
Y=200 &
\end{align*}
\]

K=20 (5 segments are shown here).

- Position is also used as part of the feature vector.
- Notice that the large background is broken.

Pros
- Very simple method.
- Converges to a local minimum.

Cons
- Memory-intensive.
- Need to pick K.
- Sensitive to initialization.
- Sensitive to outliers.
- Finds “spherical” clusters.

Segmentation by Mean Shift

- An advanced and versatile technique for clustering-based segmentation.


Mean shift algorithm (cont.)

- The mean shift algorithm seeks modes or local maxima of density in the feature space.

Mean shift algorithm (cont.)
Mean shift algorithm (cont.)

Mean shift algorithm (cont.)

Mean shift algorithm (cont.)

Mean shift algorithm (cont.)
Mean shift algorithm (cont.)

- 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 Fundamentals

\[ P(x) = \frac{1}{n} \sum_{i=1}^{n} K(x-x_i) \]

Estimate the density at a point \( x \) from a number of sample data points \( x_1, \ldots, x_n \).

- Epanechnikov Kernel
  \[ K_e(x) = \begin{cases} \frac{3}{4} \left( \frac{h}{\|x\|} \right)^2 - \frac{3}{4} \left( \frac{h}{\|x\|} \right)^3 & \text{if } \|x\| \leq h \\ 0 & \text{otherwise} \end{cases} \]

- Uniform Kernel
  \[ K_u(x) = \begin{cases} \frac{1}{n} & \text{if } \|x\| \leq h \\ 0 & \text{otherwise} \end{cases} \]

- Normal Kernel
  \[ K_n(x) = \frac{1}{h \sqrt{2\pi} \|x\|} \exp \left( -\frac{1}{2} \frac{\|x\|^2}{h^2} \right) \]

Mean Shift Fundamentals (cont.)

Give up estimating the PDF! Estimate ONLY the gradient.

Using the Kernel form:

\[ K(x-x) = \frac{1}{h} \left( \frac{x-x}{h} \right)^2 \]

We get:

\[ \nabla P(x) = \frac{1}{n} \sum_{i=1}^{n} \nabla K(x-x_i) \]

\[ \nabla P(x) = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{x-x_i}{h} \right)^2 \sum_{g \in \mathcal{G}} g \]

Simple Mean Shift procedure:
- Compute mean shift vector
- Translate the Kernel window by \( m(x) \)

Mean shift clustering/segmentation

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

Mean shift segmentation results (cont.)

Mean shift pros and cons

• Pros
  - Does not assume spherical clusters.
  - Just a single parameter (window size).
  - Finds variable number of modes.
  - Robust to outliers.

• Cons
  - Output depends on window size.
  - Computationally expensive.
  - Does not scale well with dimension of feature space.

Images as graphs

• Node for every pixel.
• Edge between every pair of pixels (or every pair of "sufficiently close" pixels).
• Each edge is weighted by the affinity or similarity of the two nodes.

Segmentation by graph partitioning

– Cut the graph into segments.
– Delete links that cross between segments.
– Easiest to break links that have low affinity.
  • similar pixels should be in the same segments.
  • dissimilar pixels should be in different segments.

Source: S. Seitz

http://www.caip.rutgers.edu/~comanici/MSPAMI/imaPamiResults.html
Measuring affinity

- Each pixel is represented by a feature vector, and a distance function is defined.
- We may convert the distance between two feature vectors into an affinity with the help of a generalized Gaussian kernel:

\[ w_{ij} = \exp\left( -\frac{1}{2\sigma^2} \text{dist}(x_i, x_j)^2 \right) \]

Scale affects affinity

- Small \( \sigma \): group nearby points.
- Large \( \sigma \): group distant points.

Graph cut

- Set of edges whose removal makes a graph disconnected.
- Cost of a cut: sum of weights of cut edges.

\[ \text{cut}(A, B) = \sum_{ij \in A \cup B} w_{ij} \]

- A graph cut gives us a segmentation
  - What is a “good” graph cut and how do we find one?

Minimum cut

- We can have a segmentation by finding the minimum cut in a graph
  - Efficient algorithms exist.

- Drawback: minimum cut tends to cut off very small, isolated components.
• Consider the image as a graph \( G(V,E) \)
  – with \( V \) being the set of vertices (pixels \( i=1,...,N \)).
  – \( E \) are the edges.
  – \( W \) is the affinity matrix between pixels.
• We want to segment it into two segments:
  – segment \( A \) containing “similar pixels” and
  – segment and \( B \) containing the rest of the image.

• We allow elements associated with
  cluster \( A \) to have a continuous weight \( a_i \).
  – Large value for \( a_i \) means a strong
  connection to the cluster.
• A good cluster is one with elements having:
  – large weights \( a_i \).
  – large values between them in the affinity
  matrix \( W \).

• An objective function expressing this assumption is:
  \[
  E(a) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} a_i a_j = \begin{bmatrix} a_1 & a_2 & \ldots & a_N \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} & \ldots & W_{1N} \\ W_{21} & W_{22} & \ldots & W_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ W_{N1} & W_{N2} & \ldots & W_{NN} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = a^T W a
  \]
  
  – \( E(a) \) is a sum of products of the form:
    • \( a_i \): association of element \( i \) with the cluster.
    • \( w_{ij} \): affinity between elements \( i \) and \( j \).
    • \( a_j \): association of element \( j \) with the cluster.

• We now estimate vector \( a \) maximizing:
  \[
  E(a) = a^T W a
  \]
  subject to:
  \[
  \|a\|^2 = 1
  \]
  because scaling \( a \) scales the objective function.

  • The Lagrangian is
    \[
    J(a; \lambda) = a^T W a + \lambda (1 - a^T a)
    \]
  leading to the solution:
    \[
    W a = \lambda a
    \]
  which is an eigenvector of \( W \). The one maximizing \( J \)
  corresponds to the largest eigenvalue of \( W \).

• Vector \( a \) is further thresholded
  – Elements of \( a \) over the threshold belong to the
    cluster.
  – Elements of \( a \) below the threshold are not
    associated with the cluster.
• More \((M)\) segments may be obtained by
  – Recursively clustering the pixels associated with
    small values of vector \( a \), or
  – Computing the first \( M \) eigenvectors of \( W \) and
    grouping their \( M \)-dimensional features (e.g. by K-
    means).
Average Association

- The presented technique may be reformulated in terms of an association between the elements of two clusters to be maximized:

\[
\frac{\text{assoc}(A, A)}{|A|} + \frac{\text{assoc}(B, B)}{|B|}, \quad \text{assoc}(A, A) = \sum_{i \in A, j \in d} w_{ij}
\]

yielding the solution of the eigenvector corresponding to the largest eigenvalue of \( W \):

\[
Wa = \lambda a
\]

Average Cut

- A similar approach minimizes the sum of edges to be cut in order to form two segments:

\[
\frac{\text{cut}(A, B)}{|A|} + \frac{\text{cut}(A, B)}{|B|}, \quad \text{cut}(A, B) = \sum_{i \in A, j \in d} w_{ij}
\]

Average Cut (cont.)

- The associated cost function to be minimized, with respect to \( a \), is:

\[
E(a) = \sum_{i=1}^{N} \sum_{j=1}^{N} (a_i - a_j)^2 w_{ij}
\]

- If pixels \( i \) and \( j \) are similar, that is \( w_{ij} \) is large, then if they do not belong to the same cluster, \( E(a) \) is heavily penalized.
- If the pixels are not similar, the energy is not affected as \( w_{ij} \) approaches zero.

Average Cut (cont.)

To facilitate the proof let us define the diagonal matrix \( D \) with elements the sum of weights arriving at the \( i \)-th pixel:

\[
E(a) = \sum_{i=1}^{N} \sum_{j=1}^{N} (a_i - a_j)^2 w_{ij}
\]

\[
= \sum_{i=1}^{N} a_i^2 \sum_{j=1}^{N} w_{ij} + \sum_{i=1}^{N} a_i^2 \sum_{j=1}^{N} w_{ij} - 2 \sum_{i=1}^{N} a_i \sum_{j=1}^{N} a_j w_{ij}
\]

\[
= \sum_{i=1}^{N} a_i^2 d_{ii} + \sum_{i=1}^{N} a_i^2 d_{jj} - 2 \sum_{i=1}^{N} a_i \sum_{j=1}^{N} a_j w_{ij}
\]

\[
= 2a^T (D-W)a
\]

Average Cut (cont.)

- Estimating vector \( a \) maximizing:

\[
E(a) = 2a^T (D-W)a
\]

subject to:

\[
a^T a = \|a\|^2 = 1
\]

yields the solution of the eigenvector corresponding to the (second) smallest eigenvalue of \( D-W \):

\[
(D-W)a = \lambda a
\]

Normalized cut

- Average association tends to find ‘tight” clusters in the graph.
- Average cut tends to keep a balance but it is not always guaranteed that that the two partitions will have a tight group similarity.
- The normalized cut (Ncut) tries to keep a balance between finding clustering (tight clusters) and segmentation (split clusters).
- It may be shown that both of the previous methods are approximations of Ncut.

J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI 2000
Normalized cut (cont.)

- Ncut is defined as:
  \[ Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \]

- The cost of the cut is a small fraction of the total affinity in each group.
- Minimization is NP-hard. An approximate solution may be found as in the previous cases.

Create vector \( a \) having components the weights associating each pixel with a cluster.

\[
a_i = \begin{cases} 
\frac{1}{assoc(A, V)} & \text{if } i \in A \\
-\frac{1}{assoc(B, V)} & \text{if } i \in B 
\end{cases}
\]

- Each \( a_i \) is a cluster indicator of the corresponding pixel.

As before, we may show that:

\[
a^T(D-W)a = \frac{1}{2} \sum_{i,j} (a_i - a_j)^2 w_{ij} = \sum_{i,j} (a_i - a_j)^2 w_{ij} \]

\[
= \sum_{i<j} \left( \frac{1}{assoc(A, V)} + \frac{1}{assoc(B, V)} \right)^2 cut(A, B)
\]

Also,

\[
a^T Da = \sum_{i,j} a_i^2 d_{ij} + \sum_{i,j} a_i a_j d_{ij}
\]

\[
= \frac{1}{assoc(A, V)} assoc(A, V) + \frac{1}{assoc(B, V)} assoc(B, V)
\]

We now combine

\[
a^T Da = \frac{1}{assoc(A, V)} + \frac{1}{assoc(B, V)} \quad \text{with}
\]

\[
a^T(D-W)a \propto \left( \frac{1}{assoc(A, V)} + \frac{1}{assoc(B, V)} \right)^2 cut(A, B)
\]

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

It turns out that the Ncut objective function is equivalent to minimizing:

\[
Ncut(A, B) = \frac{a^T(D-W)a}{a^T Da}
\]

or minimizing:

\[
E(a) = a^T(D-W)a + \lambda a^T Da
\]

which corresponds to finding the eigenvector corresponding to the (second) smallest eigenvalue of

\[ D^{-1/2}(D-W)D^{-1/2} \]
Remember the Average Cut minimization problem is equivalent to minimizing:

\[ E(a) = a^T (D - W) a \quad \text{subject to} \quad a^T a = 1 \]

The Ncut problem is equivalent to minimizing:

\[ E(a) = a^T (D - W) a \quad \text{subject to} \quad a^T Da = 1 \]

The larger the value of \( D_{ii} \), the more important is the \( i \)-th sample.

The magnitudes of the eigenvalues provide a hint for the number of clusters.

• Superpixels.
• Unsupervised bottom-up process.

Using texture features for segmentation

• How to segment images that are a “mosaic of textures”?

• Convolve image with a bank of filters.
Using texture features for segmentation (cont.)

• Find textons by clustering vectors of filter bank outputs.

Image

Texton map


C. Nikou – École Doctorale, Strasbourg 6-10 June 2011

Using texture features for segmentation (cont.)

• The final texture feature is a texton histogram computed over image windows at some “local scale”.


C. Nikou – École Doctorale, Strasbourg 6-10 June 2011

Pitfall of texture features

• Possible solution: check for “intervening contours” when computing connection weights.


C. Nikou – École Doctorale, Strasbourg 6-10 June 2011

Example Results

Results: Berkeley Segmentation Engine

http://www.cs.berkeley.edu/~fowlkes/BSE/

C. Nikou – École Doctorale, Strasbourg 6-10 June 2011

Normalized cuts: Pros and cons

• Pros
  – Generic framework, can be used with many different features and affinity formulations.

• Cons
  – High storage requirement and time complexity.
  – Bias towards partitioning into equal segments.
Segments as primitives for recognition?

Multiple segmentations.


Object detection and segmentation

Segmentation energy:

\[ E(L) = \sum \log(P(l_i | \text{class})) + \alpha \sum_{i,j \in N} \delta(l_i \neq l_j) \]


Top-down segmentation


Top-down segmentation (cont.)