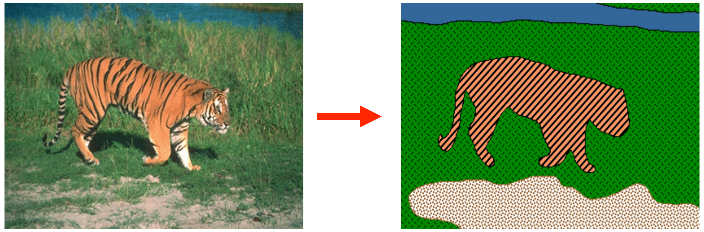
BIL 717
Image Processing

Erkut Erdem
Dept. of Computer Engineering
Hacettepe University

Clustering-based Image Segmentation

Image segmentation

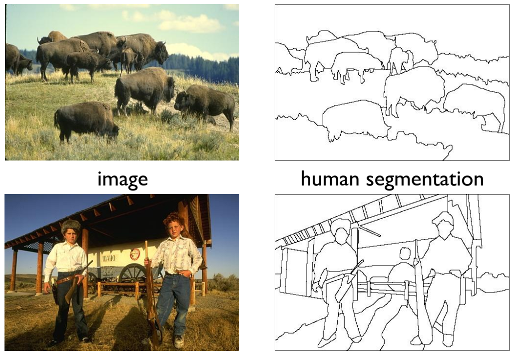
- Goal: identify groups of pixels that go together



Slide credit: S. Seitz, K. Grauman

The goals of segmentation

- Separate image into coherent "objects"



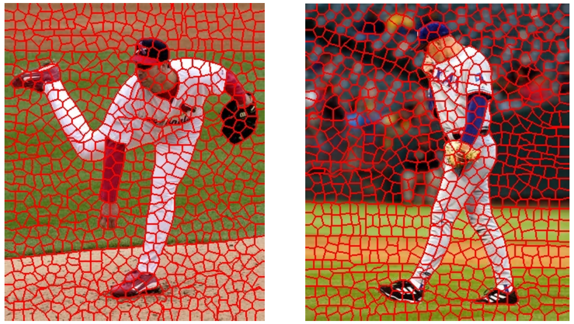
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Slide credit: S. Lazebnik

The goals of segmentation

- Separate image into coherent "objects"
- Group together similar-looking pixels for efficiency of further processing

"superpixels"



X. Ren and J. Malik. Learning a classification model for segmentation. ICCV 2003.

Slide credit: S. Lazebnik

Segmentation

- Compact representation for image data in terms of a set of components
- Components share “common” visual properties
- Properties can be defined at different level of abstractions

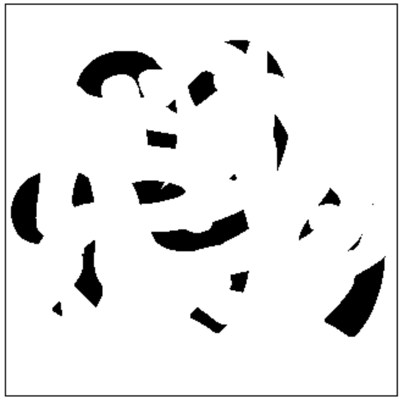
Slide credit: Fei-Fei Li

What is segmentation?

- Clustering image elements that “belong together”
  - Partitioning
    - Divide into regions/sequences with coherent internal properties
  - Grouping
    - Identify sets of coherent tokens in image

Slide credit: Fei-Fei Li

Segmentation is a global process



What are the occluded numbers?

Slide credit: B. Freeman and A. Torralba

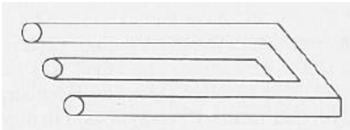
Segmentation is a global process

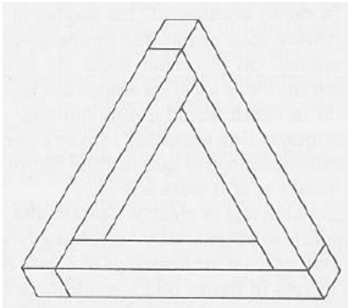
What are the occluded numbers?

Occlusion is an important cue in grouping.

Slide credit: B. Freeman and A. Torralba

… but not too global



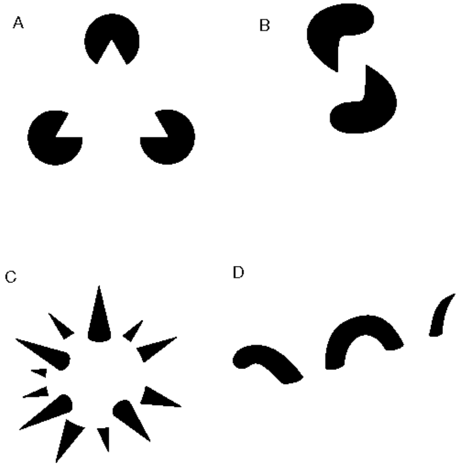


Slide credit: B. Freeman and A. Torralba

Magritte, 1957

Slide credit: B. Freeman and A. Torralba

Groupings by Invisible Completions



* Images from Steve Lehar's Gestalt papers

Slide credit: B. Freeman and A. Torralba

Groupings by Invisible Completions

1970s: R. C. James

Slide credit: B. Freeman and A. Torralba

Groupings by Invisible Completions

2000s: Bev Doolittle

Slide credit: B. Freeman and A. Torralba

Perceptual organization

"…the processes by which the bits and pieces of visual information that are available in the retinal image are structured into the larger units of perceived objects and their interrelations"

Stephen E. Palmer, *Vision Science, 1999*

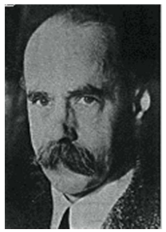
Slide credit: B. Freeman and A. Torralba

Gestalt Psychology

- German: *Gestalt* - "form" or "whole"
- Berlin School, early 20th century
  - Kurt Koffka, Max Wertheimer, and Wolfgang Köhler
- 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)

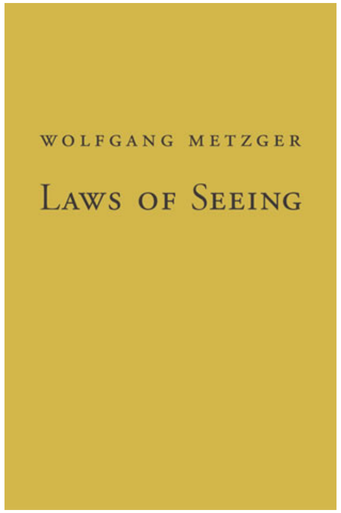
"I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees."

Max Wertheimer (1880-1943)

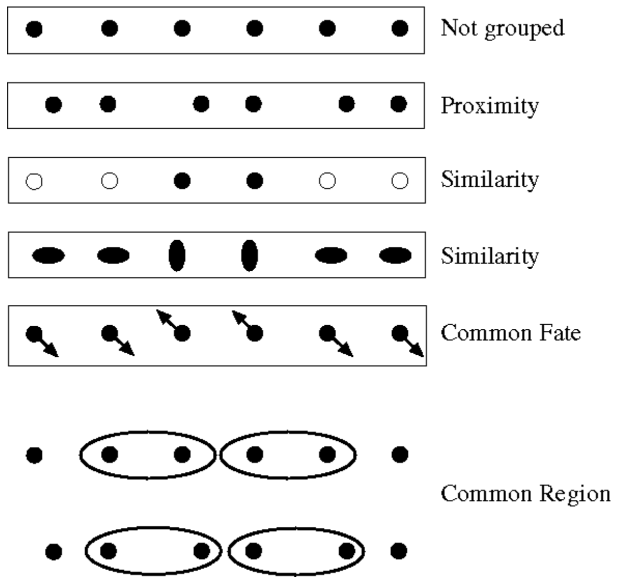


Slide credit: J. Hays and Fei-Fei Li

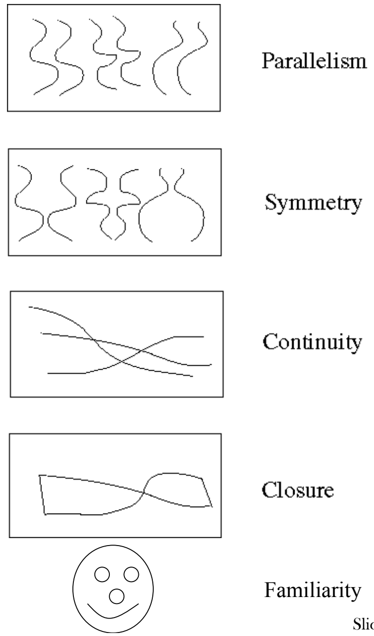
Gestalt Psychology



Laws of Seeing, Wolfgang Metzger, 1936 (English translation by Lothar Spillmann, MIT Press, 2006)



Slide credit: B. Freeman and A. Torralba

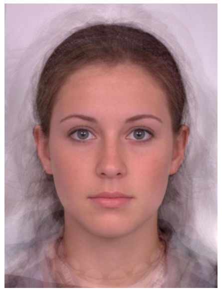


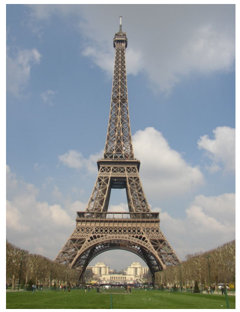
Slide credit: B. Freeman and A. Torralba

Similarity

http://chicagoist.com/attachments/chicagoist_alicia/GEESE.jpg, http://wwwdelivery.superstock.com/WI/223/1532/PreviewComp/SuperStock_1532R-0831.jpg Slide credit: K. Grauman

Symmetry





http://seedmagazine.com/news/2006/10/beauty_is_in_the_processingtim.php

Slide credit: K. Grauman

Common fate

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

Slide credit: K. Grauman

Proximity

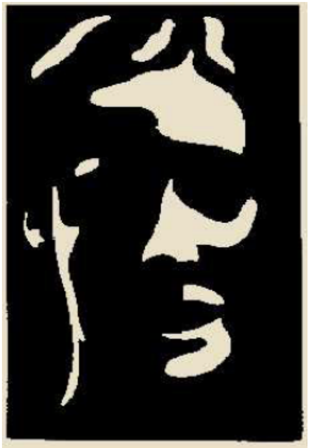
http://www.capital.edu/Resources/Images/outside6_035.jpg

Slide credit: K. Grauman

Familiarity

Slide credit: B. Freeman and A. Torralba

Familiarity



Slide credit: B. Freeman and A. Torralba

Gestalt cues

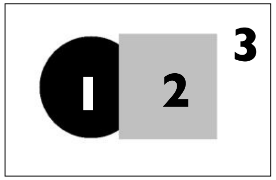
- Good intuition and basic principles for grouping
- Basis for many ideas in segmentation and occlusion reasoning
- Some (e.g., symmetry) are difficult to implement in practice

Slide credit: J. Hays

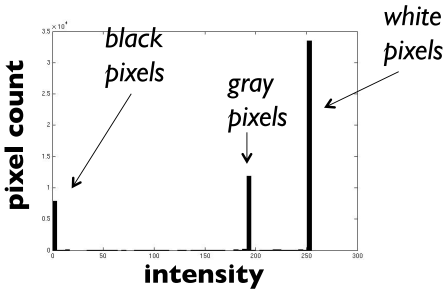
Segmentation methods

- Segmentation as clustering
  - K-means clustering
  - Mean-shift segmentation
- Graph-theoretic segmentation
  - Min cut
  - Normalized cuts

Image segmentation: toy example

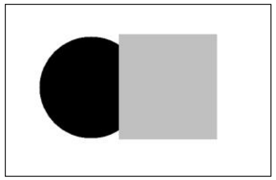


input image

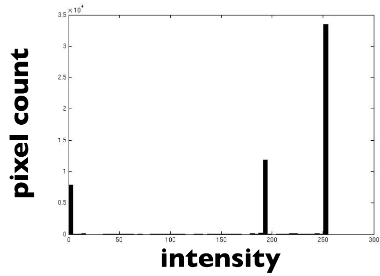


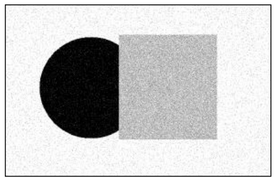
- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., *segment* the image based on the intensity feature.
- What if the image isn't quite so simple?

Slide credit: K. Grauman

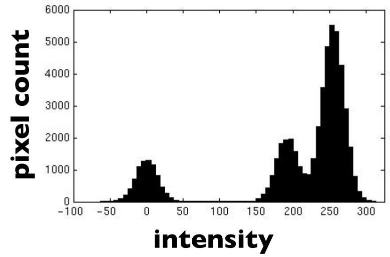


input image

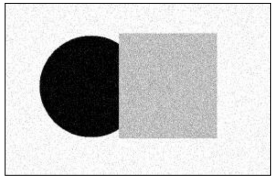




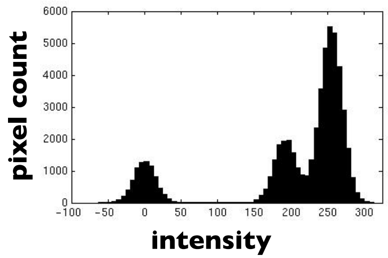
input image



Slide credit: K. Grauman

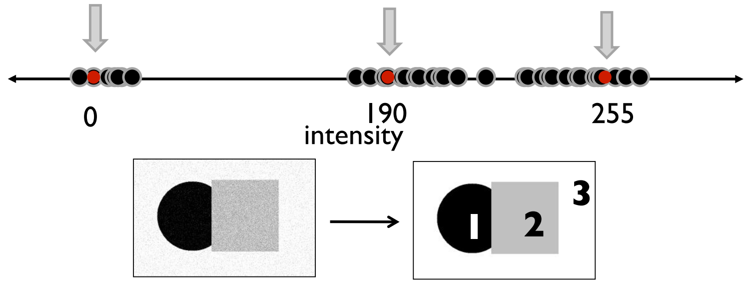


input image



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

Slide credit: K. Grauman



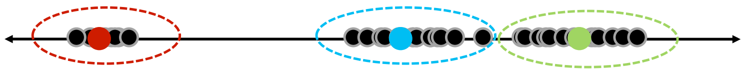
- Goal: choose three "centers" as the representative intensities, and label every pixel according to which of these centers it is nearest to.
- Best cluster centers are those that minimize SSD between all points and their nearest cluster center c_i:

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

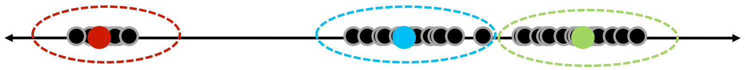
Slide credit: K. Grauman

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.



Slide credit: K. Grauman

Segmentation as clustering

- Cluster together (pixels, tokens, etc.) that belong together…
- Agglomerative clustering
  - attach closest to cluster it is closest to – repeat
- Divisive clustering
  - split cluster along best boundary – repeat
- Dendrograms
  - yield a picture of output as clustering process continues

Slide credit: B. Freeman

Greedy Clustering Algorithms

Algorithm 15.3: Agglomerative clustering, or clustering by merging

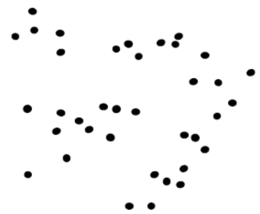
```
Make each point a separate cluster
Until the clustering is satisfactory
    Merge the two clusters with the
      smallest inter-cluster distance
end
```

Algorithm 15.4: Divisive clustering, or clustering by splitting

```
Construct a single cluster containing all points
Until the clustering is satisfactory
    Split the cluster that yields the two
      components with the largest inter-cluster distance
end
```

Slide credit: B. Freeman

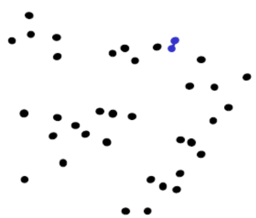
Agglomerative clustering



1. Say "Every point is its own cluster"

Slide credit: D. Hoiem

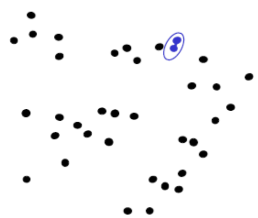
Agglomerative clustering



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters

Slide credit: D. Hoiem

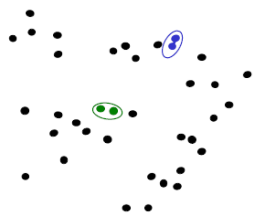
Agglomerative clustering



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster

Slide credit: D. Hoiem

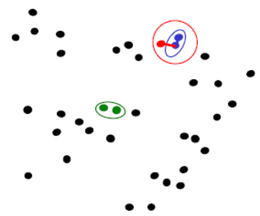
Agglomerative clustering



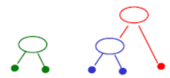
1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat

Slide credit: D. Hoiem

Agglomerative clustering



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat



Slide credit: D. Hoiem

Common similarity/distance measures

- P-norms
  - City Block (L1)
  - Euclidean (L2)
  - L-infinity

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

$$\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|$$

$$\|\boldsymbol{x}\| := \sqrt{x_1^2 + \cdots + x_n^2}$$

$$\|\mathbf{x}\|_\infty := \max\left(|x_1|, \ldots, |x_n|\right)$$

Here x_i is the distance btw. two points

- Mahalanobis
  - Scaled Euclidean

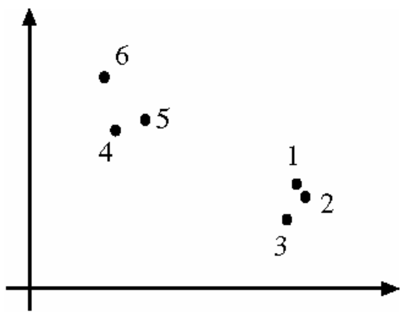
$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{\sigma_i^2}}$$

- Cosine distance

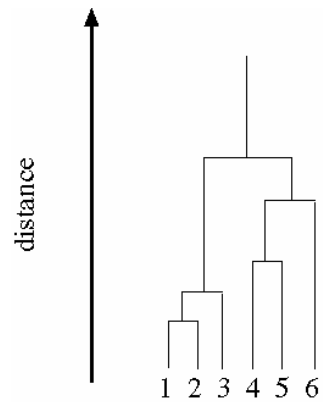
$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Slide credit: D. Hoiem

Dendograms



Data set



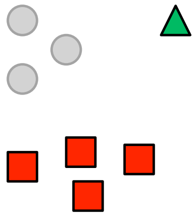
Dendogram formed by agglomerative clustering using single-link clustering.

Slide credit: B. Freeman

Agglomerative clustering

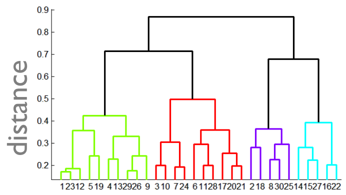
How to define cluster similarity?

- Average distance between points, maximum distance, minimum distance
- Distance between means or medoids



How many clusters?

- Clustering creates a dendrogram (a tree)
- Threshold based on max number of clusters or based on distance between merges



Slide credit: D. Hoiem

Agglomerative clustering

Good

- Simple to implement, widespread application
- Clusters have adaptive shapes
- Provides a hierarchy of clusters

Bad

- May have imbalanced clusters
- Still have to choose number of clusters or threshold
- Need to use an "ultrametric" to get a meaningful hierarchy

Slide credit: D. Hoiem

Segmentation methods

- Segmentation as clustering
  - K-means clustering
  - Mean-shift segmentation
- Graph-Theoretic Segmentation
  - Min cut
  - Normalized cuts

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

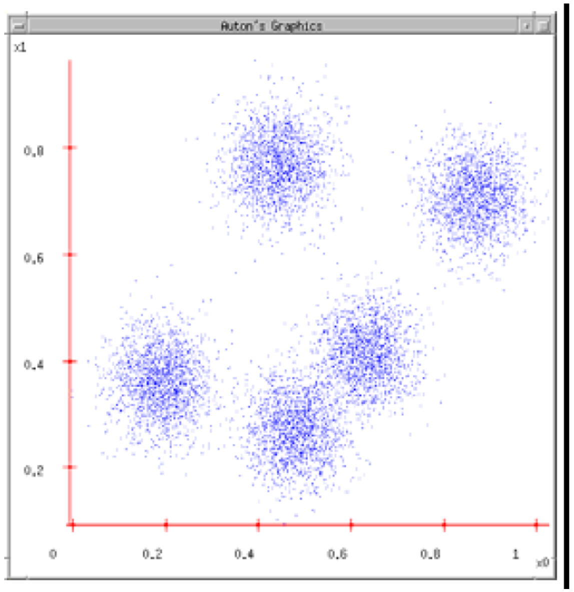
- Will always converge to *some* solution
- Can be a "local minimum"
  - does not always find the global minimum of objective function:

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

Slide credit: S. Seitz

K-means

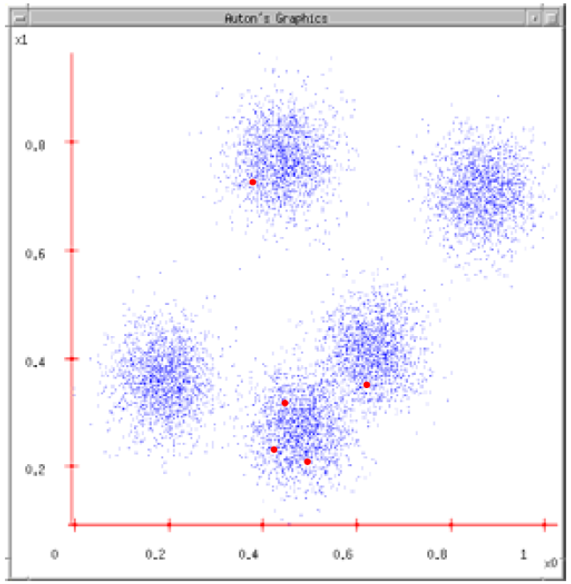
1. Ask user how many clusters they'd like. *(e.g. k=5)*



Slide credit: K Grauman, A. Moore

K-means

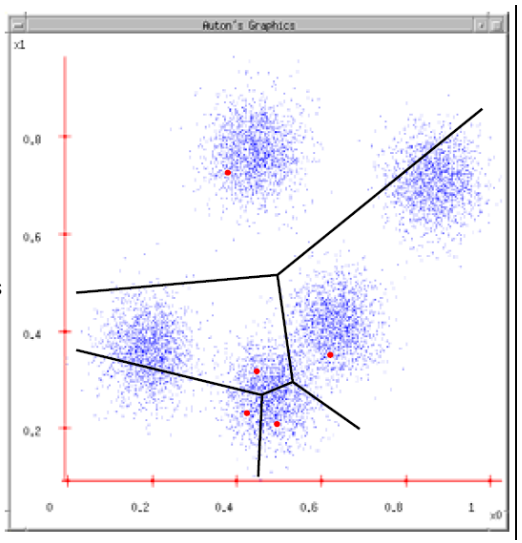
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



Slide credit: K Grauman, A. Moore

K-means

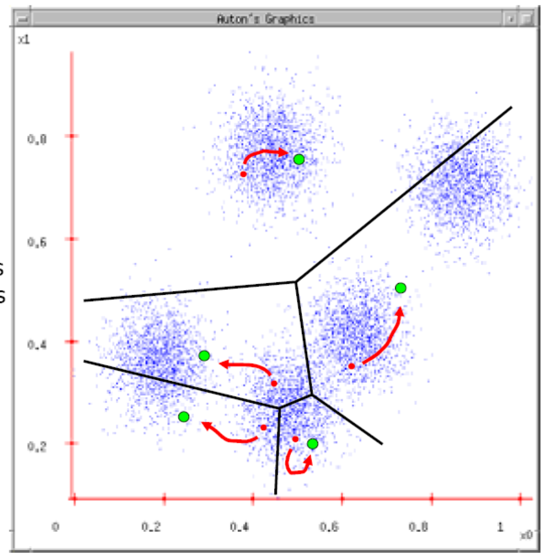
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



Slide credit: K Grauman, A. Moore

K-means

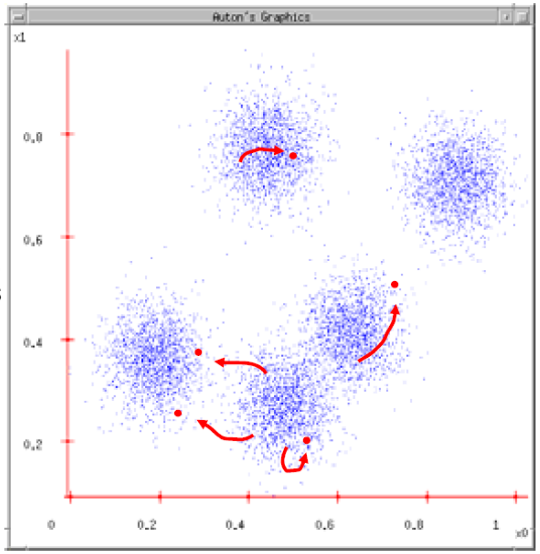
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



Slide credit: K Grauman, A. Moore

K-means

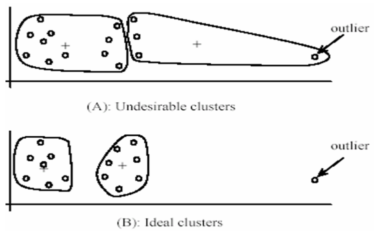
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



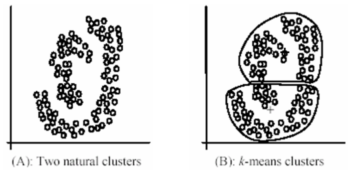
Slide credit: K Grauman, A. Moore

K-means: pros and cons

Pros
- Simple, fast to compute
- Converges to local minimum of within-cluster squared error



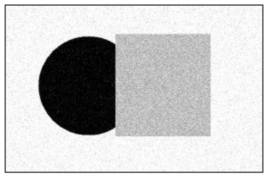
Cons/issues
- Setting k?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters
- Assuming means can be computed



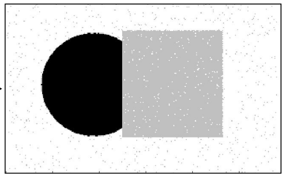
Slide credit: K Grauman

An aside: Smoothing out cluster assignments

- Assigning a cluster label per pixel may yield outliers:



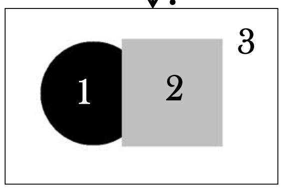
original



labeled by cluster center's intensity

?

- How to ensure they are spatially smooth?



Slide credit: K 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 similarity

Feature space: intensity value (1-d)

Slide credit: K Grauman

K=2

K=3

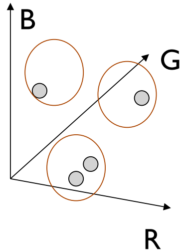
quantization of the feature space; segmentation label map

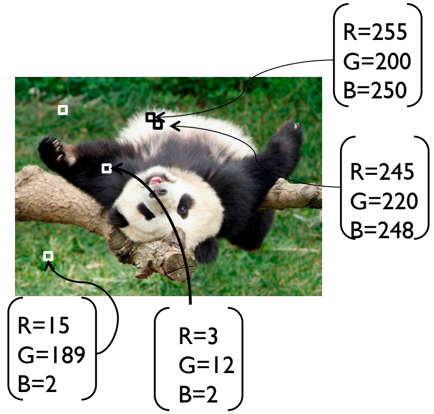
Slide credit: K Grauman

Segmentation as clustering

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

Grouping pixels based on <u>color</u> similarity





Feature space: color value (3-d)

Slide credit: K 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** similarity

Clusters based on intensity similarity don't have to be spatially coherent.

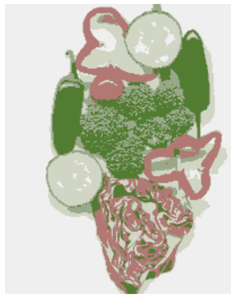
Slide credit: K Grauman

Segmentation as clustering

Image

Clusters on intensity (K=5)

Clusters on color (K=5)



K-means clustering using intensity alone and color alone

Slide credit: B. Freeman

Segmentation as clustering

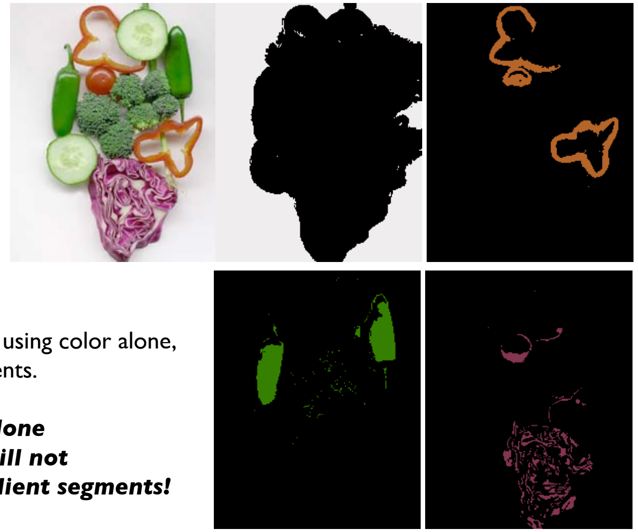
Image

Clusters on color

K-means using color alone, 11 segments

Slide credit: B. Freeman

Segmentation as clustering



K-means using color alone, 11 segments.

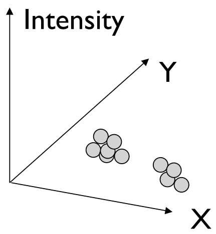
Color alone often will not yield salient segments!

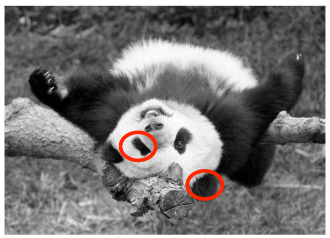
Slide credit: B. Freeman

Segmentation as clustering

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

Grouping pixels based on <u>intensity+position</u> similarity





Both regions are black, but if we also include <u>position (x,y)</u>, then we could group the two into distinct segments; way to encode both similarity & proximity.

Slide credit: K Grauman

Segmentation as clustering

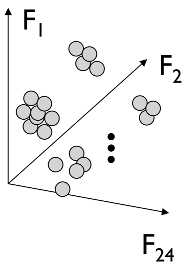
- Color, brightness, position alone are not enough to distinguish all regions…

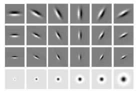
Slide credit: K Grauman

Segmentation as clustering

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

Grouping pixels based on <u>texture</u> similarity



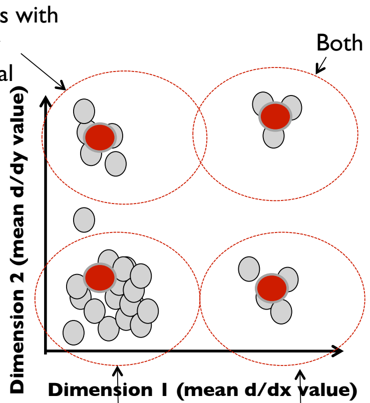


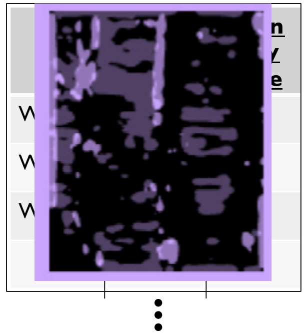
Filter bank of 24 filters

Feature space: filter bank responses (e.g., 24-d)

Slide credit: K Grauman

Texture representation example





statistics to summarize patterns in small windows

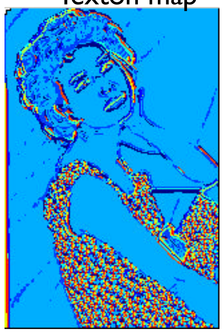
Slide credit: K Grauman

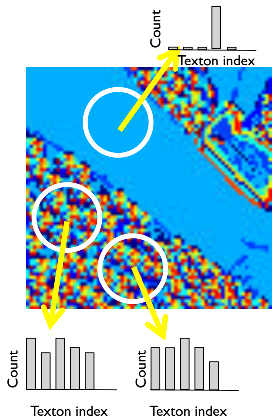
Segmentation with texture features

- Find "textons" by **clustering** vectors of filter bank outputs
- Describe texture in a window based on *texton histogram*

Image

Texton map

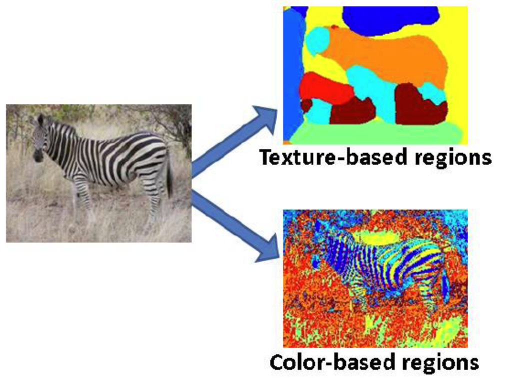




Malik, Belongie, Leung and Shi. IJCV 2001.

Slide credit: K Grauman, L. Lazebnik

Image segmentation example



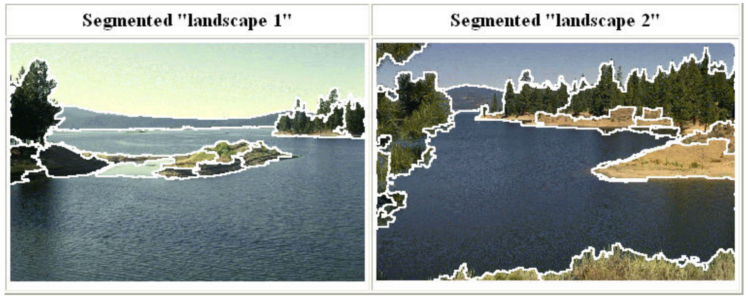
Slide credit: K Grauman

Segmentation methods

- Segmentation as clustering
  - K-means clustering
  - Mean-shift segmentation
- Graph-theoretic segmentation
  - Min cut
  - Normalized cuts
- Interactive segmentation

Mean shift clustering and segmentation

- An advanced and versatile technique for clustering-based segmentation

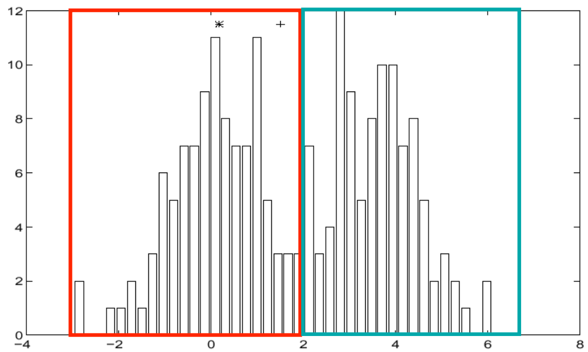


http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

Slide credit: S. Lazebnik

Finding Modes in a Histogram



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

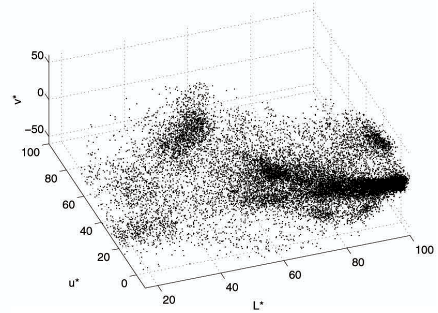
Slide credit: S. Seitz

Mean shift algorithm

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

image

Feature space
(L*u*v* color values)



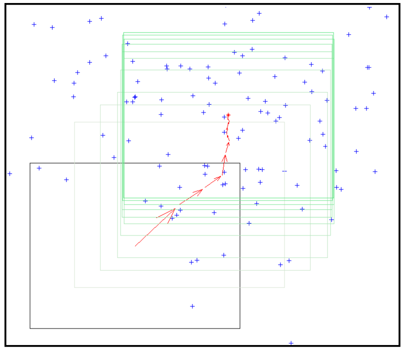
Slide credit: S. Lazebnik

Mean shift algorithm

Mean Shift Algorithm

1. Choose a search window size.
2. Choose the initial location of the search window.
3. Compute the mean location (centroid of the data) in the search window.
4. Center the search window at the mean location computed in Step 3.
5. Repeat Steps 3 and 4 until convergence.

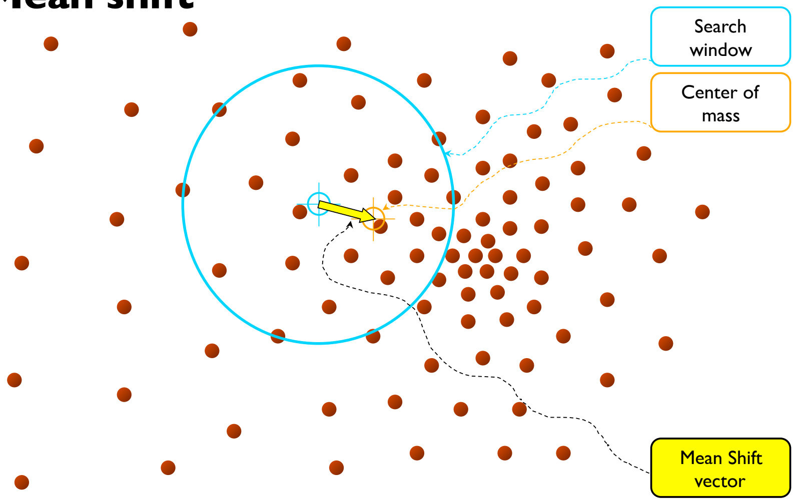
The mean shift algorithm seeks the "mode" or point of highest density of a data distribution:



Two issues:
(1) Kernel to interpolate density based on sample positions.
(2) Gradient ascent to mode.

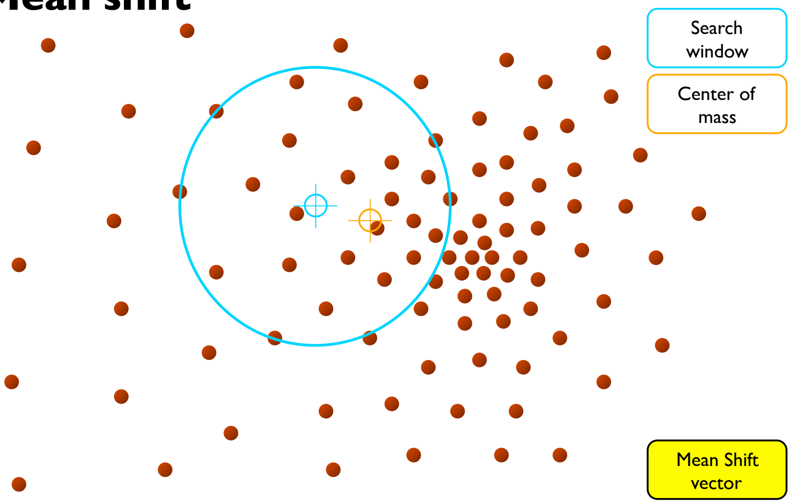
Slide credit: B. Freeman and A. Torralba

Mean shift



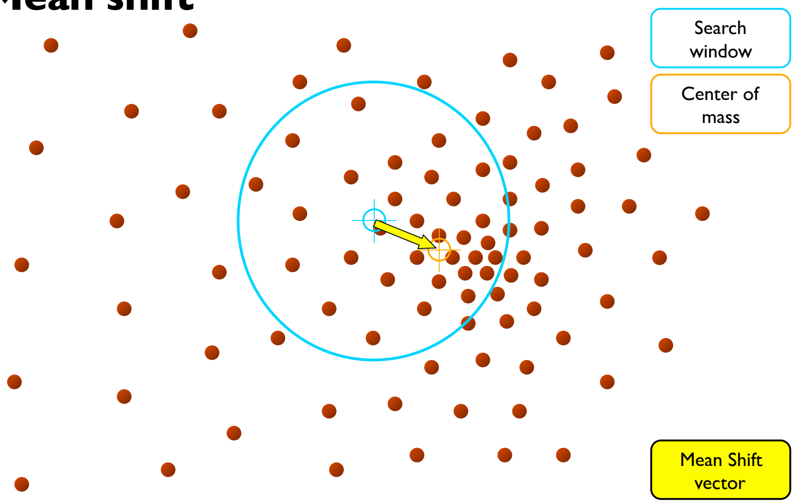
Slide credit: Y. Ukrainitz & B. Sarel

Mean shift



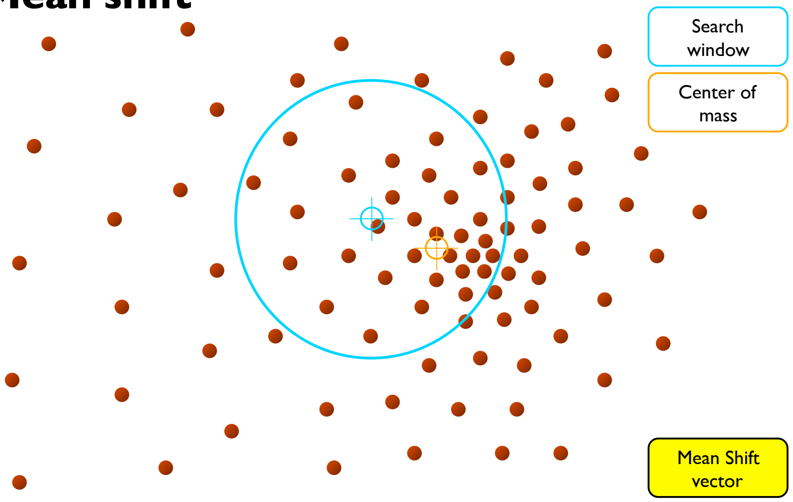
Slide credit: Y. Ukrainitz & B. Sarel

Mean shift



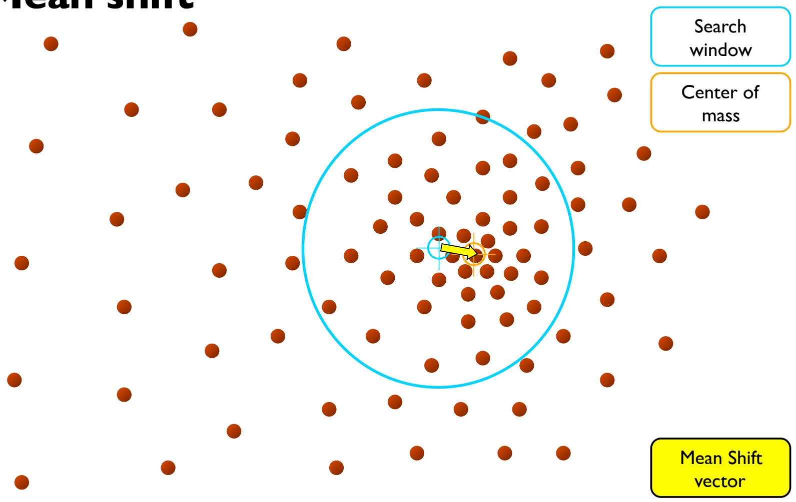
Slide credit: Y. Ukrainitz & B. Sarel

Mean shift



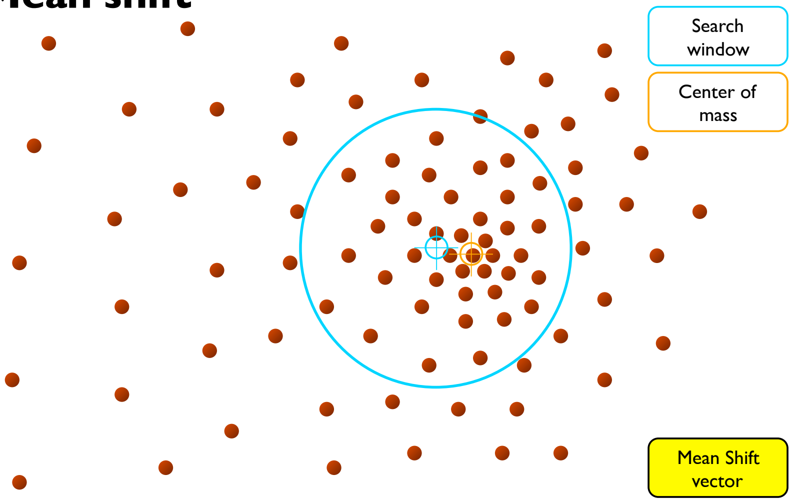
Slide credit: Y. Ukrainitz & B. Sarel

Mean shift



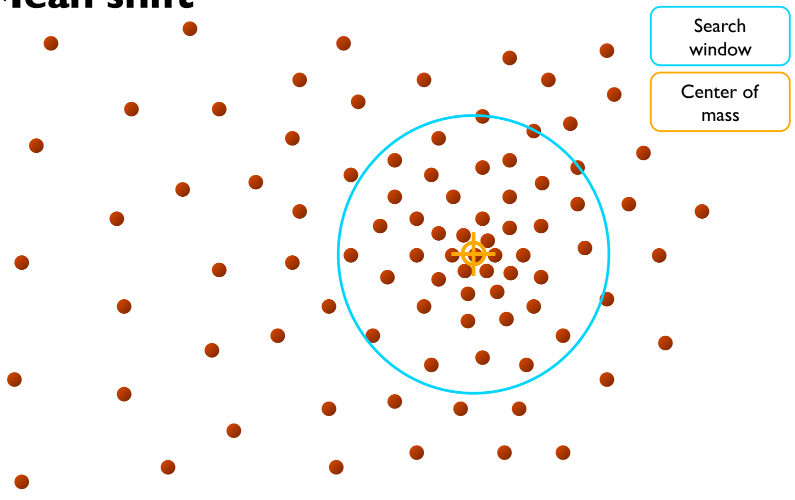
Slide credit: Y. Ukrainitz & B. Sarel

Mean shift



Slide credit: Y. Ukrainitz & B. Sarel

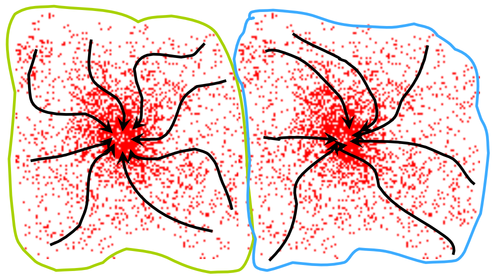
Mean shift



Slide credit: Y. Ukrainitz & B. Sarel

Mean shift clustering

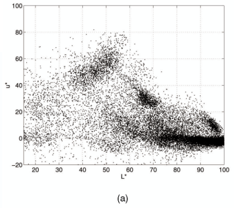
- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode



Slide credit: Y. Ukrainitz & B. Sarel

Mean shift clustering/segmentation

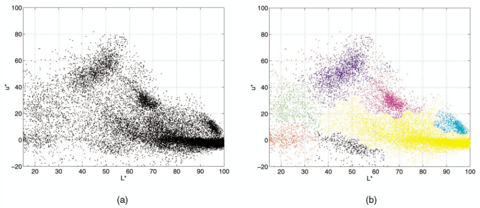
- Find features (color, gradients, texture, etc)
- Initialize windows at individual feature points
- Perform mean shift for each window until convergence
- Merge windows that end up near the same “peak” or mode

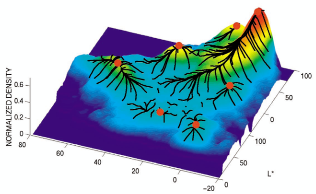


Slide credit: S. Lazebnik

Mean shift clustering/segmentation

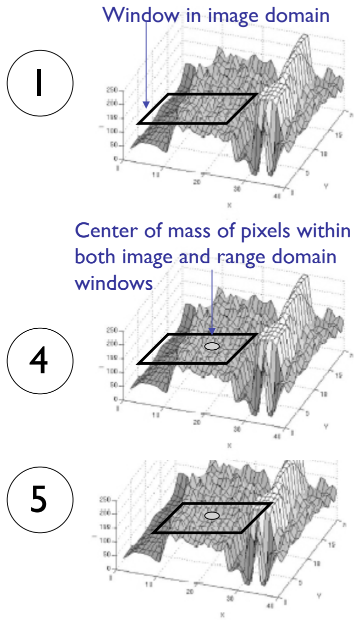
- Find features (color, gradients, texture, etc)
- Initialize windows at individual feature points
- Perform mean shift for each window until convergence
- Merge windows that end up near the same “peak” or mode

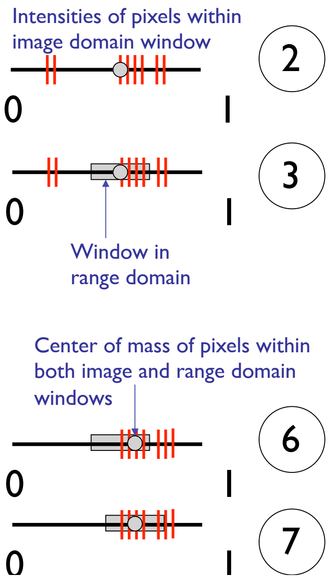




Slide credit: S. Lazebnik

Apply mean shift jointly in the image (left col.) and range (right col.) domains





Slide credit: B. Freeman and A. Torralba

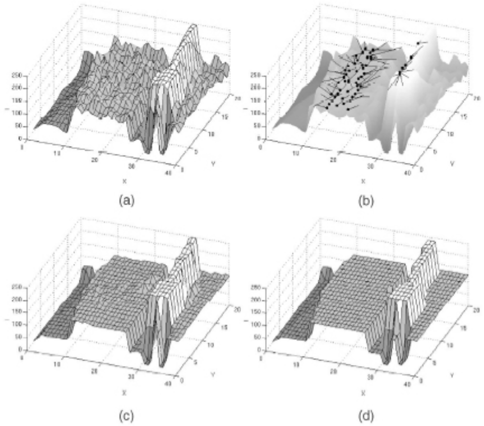
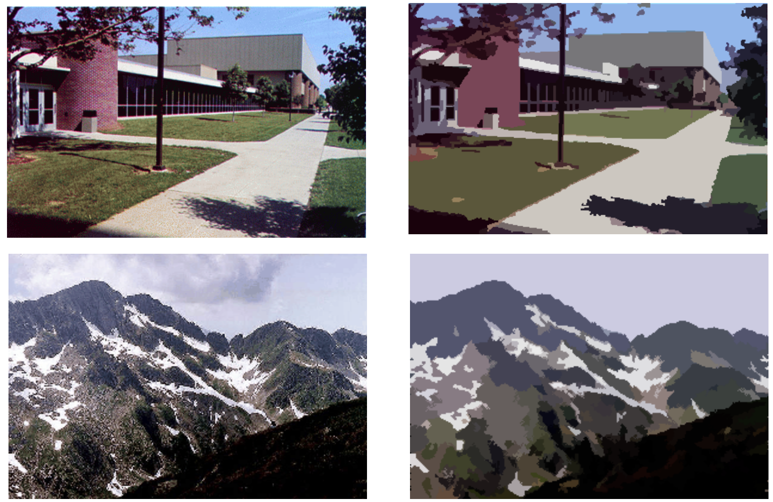


Fig. 4. Visualization of mean shift-based filtering and segmentation for gray-level data. (a) Input. (b) Mean shift paths for the pixels on the plateau and on the line. The black dots are the points of convergence. (c) Filtering result $(h_s, h_r) = (8, 4)$. (d) Segmentation result.

Comaniciu and Meer, IEEE PAMI vol. 24, no. 5, 2002

Slide credit: B. Freeman and A. Torralba

Mean shift segmentation results



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

Slide credit: S. Lazebnik

More results

Slide credit: S. Lazebnik

More results

Slide credit: S. Lazebnik

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

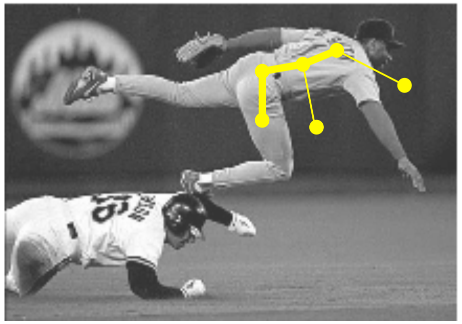
Slide credit: S. Lazebnik

Segmentation methods

- Segmentation as clustering
  - K-means clustering
  - Mean-shift segmentation
- Graph-theoretic segmentation
  - Min cut
  - Normalized cuts

Graph-Theoretic Image Segmentation

Build a weighted graph G=(V,E) from image



V: image pixels

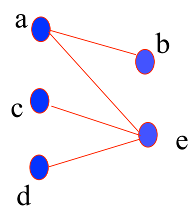
E: connections between pairs of nearby pixels

W_{ij} : probability that i &j belong to the same region

Segmentation = graph partition

Slide credit: B. Freeman and A. Torralba

Graphs Representations



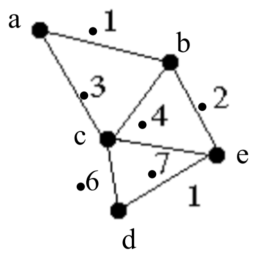
$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Adjacency Matrix

Slide credit: B. Freeman and A. Torralba

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

A Weighted Graph and its Representation



Affinity Matrix

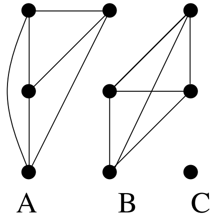
$$W = \begin{bmatrix} 1 & .1 & .3 & 0 & 0 \\ .1 & 1 & .4 & 0 & .2 \\ .3 & .4 & 1 & .6 & .7 \\ 0 & 0 & .6 & 1 & 1 \\ 0 & .2 & .7 & 1 & 1 \end{bmatrix}$$

W_{ij} : probability that i &j belong to the same region

Slide credit: B. Freeman and A. Torralba

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Segmentation by graph partitioning



- Break 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

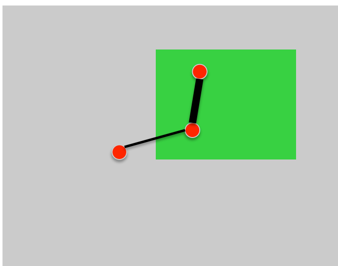
Slide credit: S. Seitz

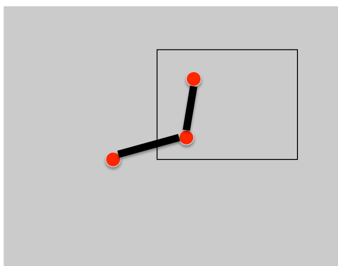
Affinity between pixels

Similarities among pixel descriptors

$W_{ij} = \exp(-|| z_i - z_j ||^2 / \sigma^2)$

σ = Scale factor… it will hunt us later





Slide credit: B. Freeman and A. Torralba

Affinity between pixels

Similarities among pixel descriptors

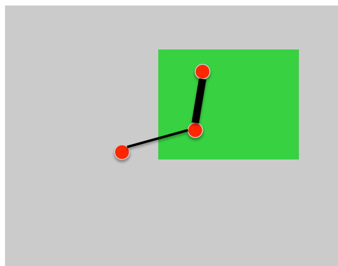
$W_{ij} = \exp(-|| z_i - z_j ||^2 / \sigma^2)$

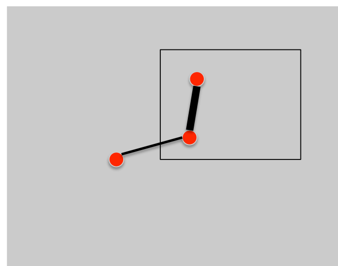
σ = Scale factor… it will hunt us later

Interleaving edges

$W_{ij} = 1 - \max_{\text{Line between i and j}} Pb$

With Pb = probability of boundary

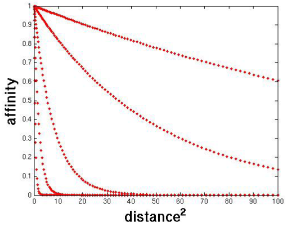


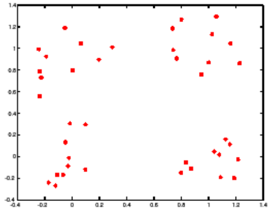


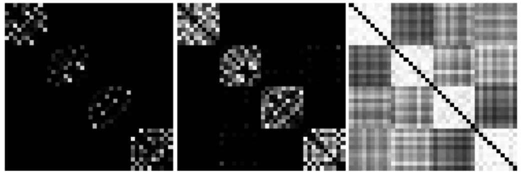
Slide credit: B. Freeman and A. Torralba

Scale affects affinity

- Small σ: group only nearby points
- Large σ: group far-away points







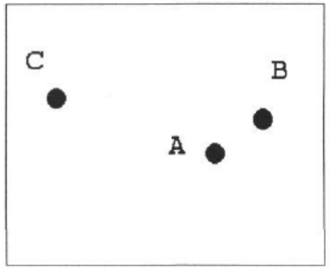
Slide credit: S. Lazebnik

Feature grouping by "relocalisation" of eigenvectors of the proximity matrix

British Machine Vision Conference, pp. 103-108, 1990

Guy L. Scott
Robotics Research Group
Department of Engineering Science
University of Oxford

H. Christopher Longuet-Higgins
University of Sussex
Falmer
Brighton



Three points in feature space

$W_{ij} = \exp(-|| z_i - z_j ||^2 / s^2)$

With an appropriate s

W=

	A	B	C
A	1.00	0.63	0.03
B	0.63	1.00	0.0
C	0.03	0.0	1.00

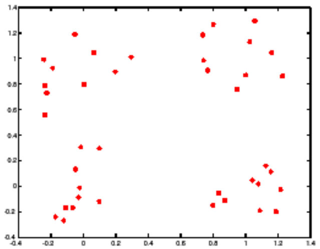
The eigenvectors of W are:

	E_1	E_2	E_3
Eigenvalues	1.63	1.00	0.37
A	-0.71	-0.01	0.71
B	-0.71	-0.05	-0.71
C	-0.04	1.00	-0.03

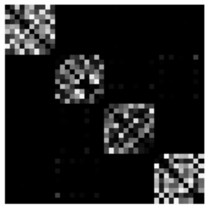
The first 2 eigenvectors group the points as desired…

Slide credit: B. Freeman and A. Torralba

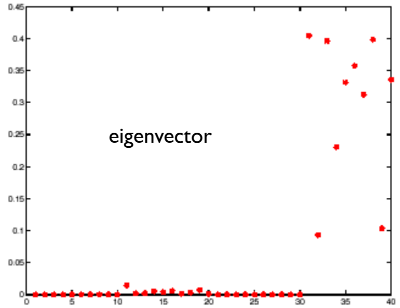
Example eigenvector



points

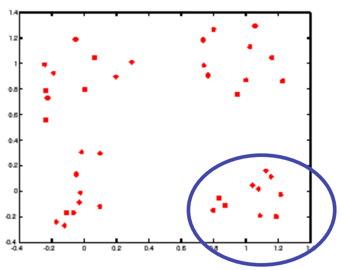


Affinity matrix

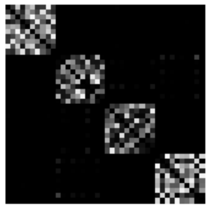


Slide credit: B. Freeman and A. Torralba

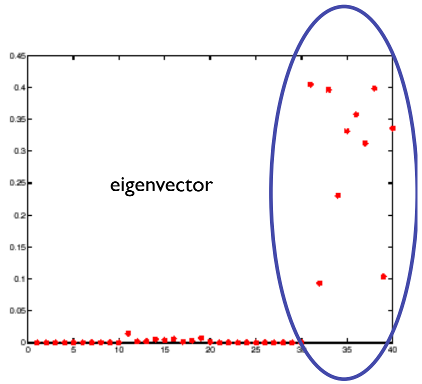
Example eigenvector



points

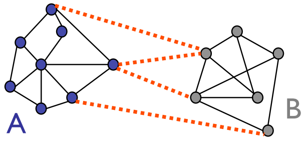


Affinity matrix



Slide credit: B. Freeman and A. Torralba

Graph cut



- Set of edges whose removal makes a graph disconnected
- Cost of a cut: sum of weights of cut edges
- A graph cut gives us a segmentation
  - What is a "good" graph cut and how do we find one?

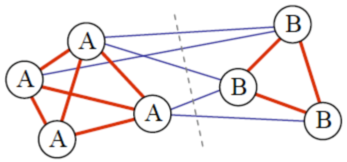
Slide credit: S. Seitz

Segmentation methods

- Segmentation as clustering
  - K-means clustering
  - Mean-shift segmentation
- Graph-theoretic segmentation
  - Min cut
  - Normalized cuts

Minimum cut

A cut of a graph *G* is the set of edges *S* such that removal of *S* from *G* disconnects *G*.



Cut: sum of the weight of the cut edges:

$$cut(\mathrm{A},\mathrm{B}) = \sum_{u \in \mathrm{A}, v \in \mathrm{B}} \mathrm{W}(u,v),$$

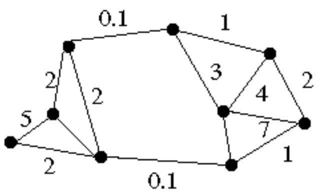
with $\mathrm{A} \cap \mathrm{B} = \varnothing$

Slide credit: B. Freeman and A. Torralba

Minimum cut

- We can do segmentation by finding the *minimum cut* in a graph
  - Efficient algorithms exist for doing this

Minimum cut example

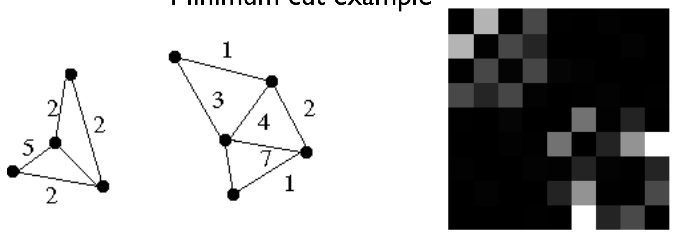


Slide credit: S. Lazebnik

Minimum cut

- We can do segmentation by finding the *minimum cut* in a graph
  - Efficient algorithms exist for doing this

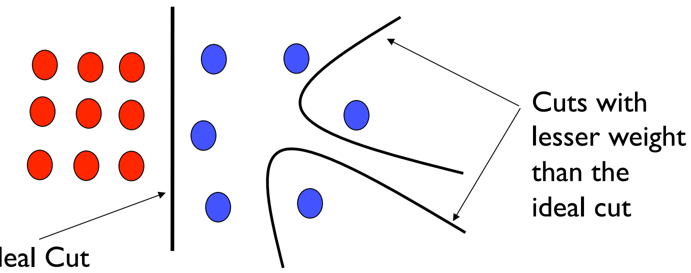
Minimum cut example



Slide credit: S. Lazebnik

Drawbacks of Minimum cut

- Weight of cut is directly proportional to the number of edges in the cut.



Slide credit: B. Freeman and A. Torralba

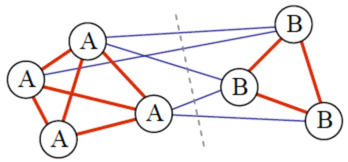
* Slide from Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Segmentation methods

- Segmentation as clustering
  - K-means clustering
  - Mean-shift segmentation
- Graph-theoretic segmentation
  - Min cut
  - Normalized cuts

Normalized cuts

Write graph as V, one cluster as A and the other as B



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

cut(A,B) is sum of weights with one end in A and one end in B

$$cut(A,B) = \sum_{u \in A, v \in B} W(u,v),$$

with $A \cap B = \emptyset$

assoc(A,V) is sum of all edges with one end in A.

$$assoc(A,B) = \sum_{u \in A, v \in B} W(u,v)$$

A and B not necessarily disjoint

J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI 2000

Slide credit: B. Freeman and A. Torralba

Normalized cut

- Let W be the adjacency matrix of the graph
- Let D be the diagonal matrix with diagonal entries
  $D(i, i) = \sum_j W(i, j)$
- Then the normalized cut cost can be written as

$$\frac{y^T (D-W) y}{y^T D y}$$

  where y is an indicator vector whose value should be 1 in the ith position if the ith feature point belongs to A and a negative constant otherwise

J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI 2000

Slide credit: S. Lazebnik

Normalized cut

- Finding the exact minimum of the normalized cut cost is NP-complete, but if we *relax* y to take on arbitrary values, then we can minimize the relaxed cost by solving the *generalized eigenvalue problem* $(D - W)y = \lambda Dy$
- The solution y is given by the generalized eigenvector corresponding to the second smallest eigenvalue
- Intitutively, the ith entry of y can be viewed as a "soft" indication of the component membership of the ith feature
  - Can use 0 or median value of the entries as the splitting point (threshold), or find threshold that minimizes the Ncut cost

J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI 2000

Slide credit: S. Lazebnik

Normalized cut algorithm

1. Given an image or image sequence, set up a weighted graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, and set the weight on the edge connecting two nodes being a measure of the similarity between the two nodes.
2. Solve $(\mathbf{D} - \mathbf{W})\boldsymbol{x} = \lambda \mathbf{D}\boldsymbol{x}$ for eigenvectors with the smallest eigenvalues.
3. Use the eigenvector with second smallest eigenvalue to bipartition the graph.
4. Decide if the current partition should be sub-divided, and recursively repartition the segmented parts if necessary.

Slide credit: B. Freeman and A. Torralba

Global optimization

- In this formulation, the segmentation becomes a global process.
- Decisions about what is a boundary are not local (as in Canny edge detector)

Slide credit: B. Freeman and A. Torralba

Boundaries of image regions defined by a number of attributes

- Brightness/color
- Texture
- Motion
- Stereoscopic depth
- Familiar configuration

[Malik]

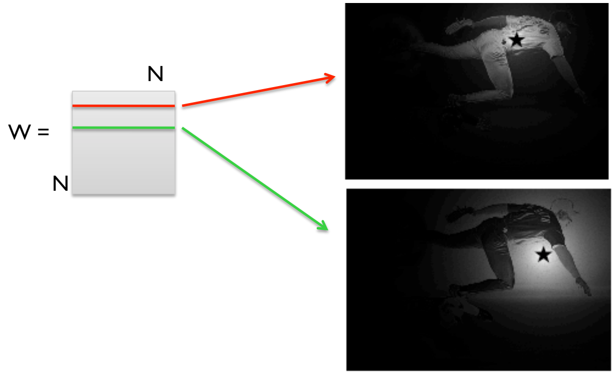
Slide credit: B. Freeman and A. Torralba

Example

Affinity:

$$w_{ij} = \underbrace{e^{\frac{-\|F_{(i)}-F_{(j)}\|_2^2}{\sigma_I}}}_{\text{brightness}} * \underbrace{\begin{cases} e^{\frac{-\|X_{(i)}-X_{(j)}\|_2^2}{\sigma_X}} & \text{if } \|X(i)-X(j)\|_2 < r \\ 0 & \text{otherwise} \end{cases}}_{\text{Location}}$$

N pixels = ncols * nrows



Slide credit: B. Freeman and A. Torralba

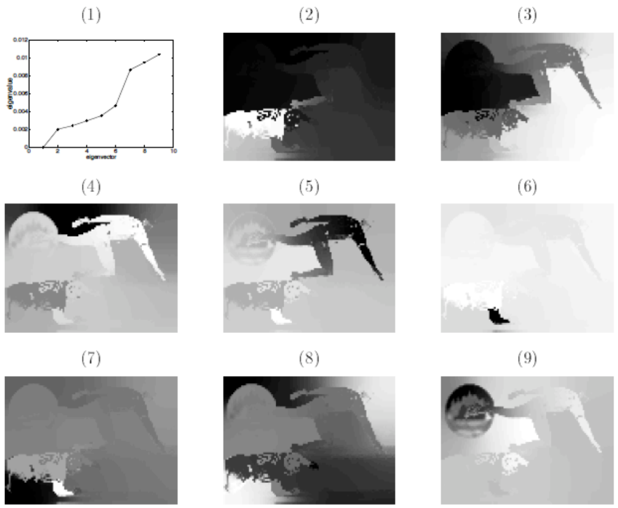
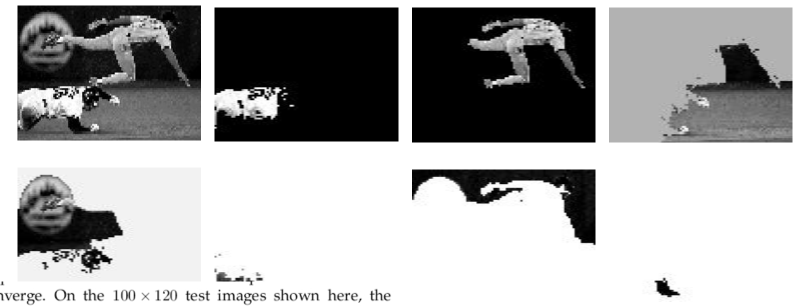


Figure 12: Subplot (1) plots the smallest eigenvectors of the generalized eigenvalue system (11). Subplot (2) - (9) shows the eigenvectors corresponding the 2nd smallest to the 9th smallest eigenvalues of the system. The eigenvectors are reshaped to be the size of the image.

Slide credit: B. Freeman and A. Torralba

Brightness Image Segmentation



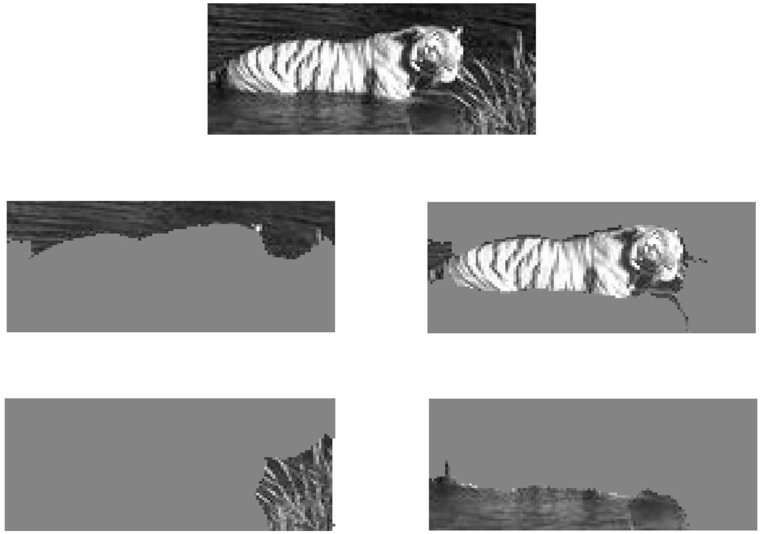
converge. On the 100 × 120 test images shown here, the normalized cut algorithm takes about 2 minutes on Intel Pentium 200MHz machines.

A multiresolution implementation can be used to reduce this running time further on larger images. In our current experiments, with this implementation, the running time on a 300 × 400 image can be reduced to about 20 seconds on Intel Pentium 300MHz machines. Furthermore, the bottleneck of the computation, a sparse matrix-vector

http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf

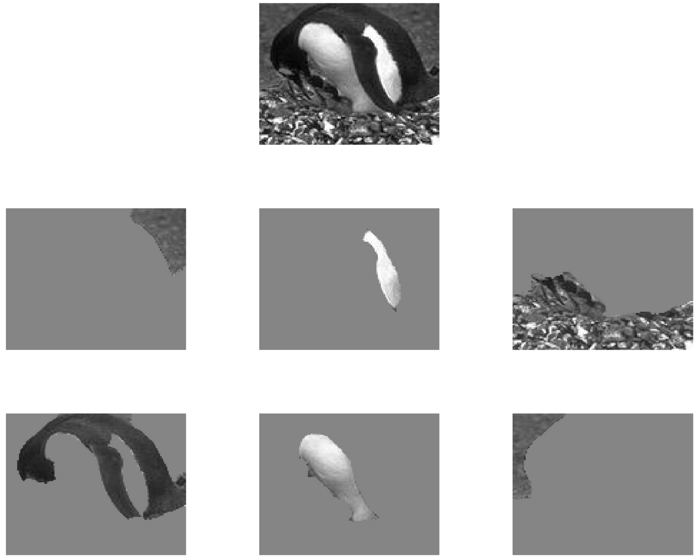
Slide credit: B. Freeman and A. Torralba

Brightness Image Segmentation



http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf

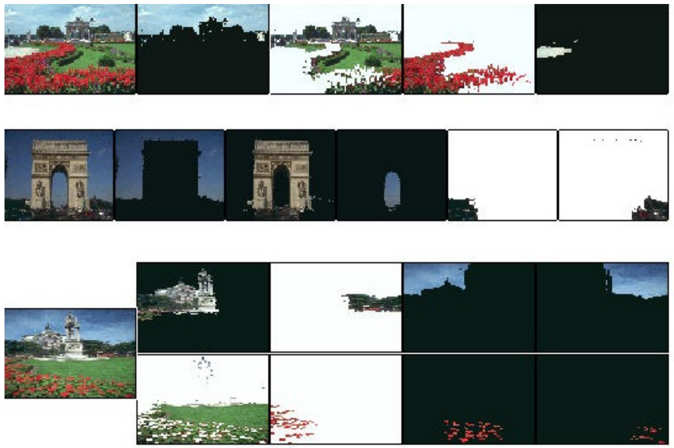
Slide credit: B. Freeman and A. Torralba



http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf

Slide credit: B. Freeman and A. Torralba

Results on color segmentation



http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf

Slide credit: B. Freeman and A. Torralba

Example results

Slide credit: S. Lazebnik

Results: Berkeley Segmentation Engine

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

Slide credit: S. Lazebnik

Normalized cuts: Pro and con

- 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

Slide credit: S. Lazebnik