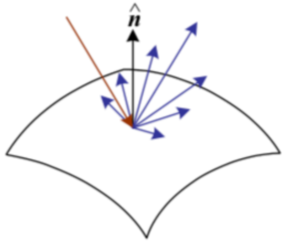


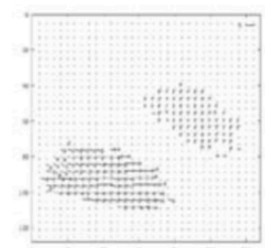
2. Image Formation

3. Image Processing

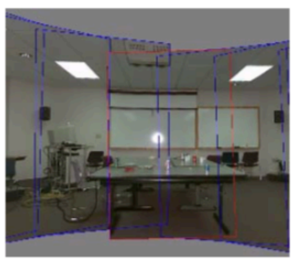
4. Features

5. Segmentation

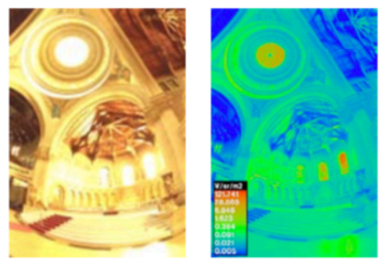
6-7. Structure from Motion



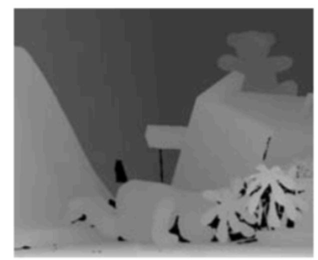
8. Motion



9. Stitching

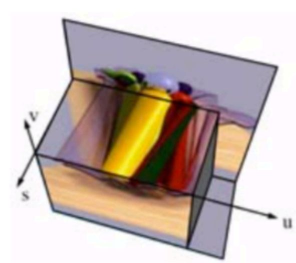


10. Computational Photography



11. Stereo

12. 3D Shape



13. Image-based Rendering

14. Recognition

Segmentation & Custering

Disclaimer: Many slides have been borrowed from Devi Parikh and Kristen Grauman, who may have borrowed some of them from others. Any time a slide did not already have a credit on it, I have credited it to Kristen. So there is a chance some of these credits are inaccurate.

Grouping in Vision

Segmentation as Clustering

Mode finding & Mean-Shift

Graph-Based Algorithms

Segments as Primitives

CNN-Based Approaches

Grouping in Vision

Segmentation as Clustering

Mode finding & Mean-Shift

Graph-Based Algorithms

Segments as Primitives

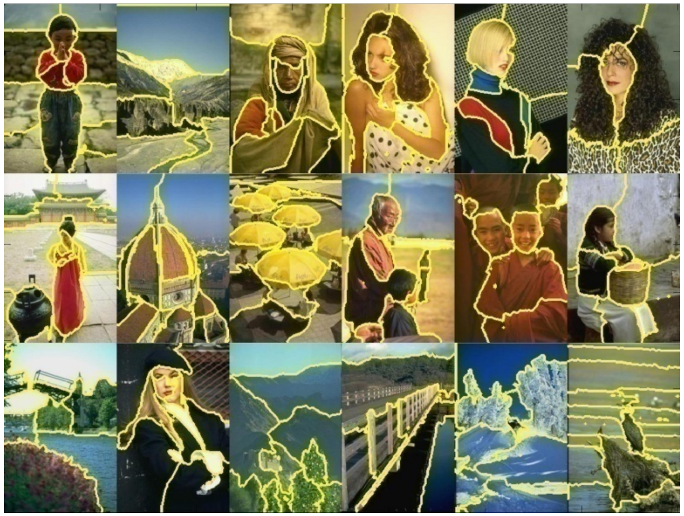
CNN-Based Approaches

Grouping in vision

- Goals:
  - Gather features that belong together
  - Obtain an intermediate representation that compactly describes key image or video parts

Slide credit: Kristen Grauman

Examples of grouping in vision



[Figure by J. Shi]

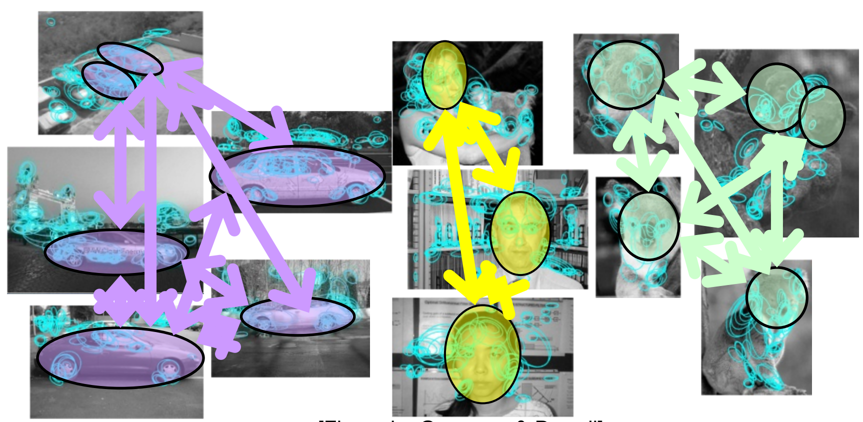
Determine image regions

[http://poseidon.csd.auth.gr/LAB_RESEARCH/Latest/imgs/SpeakDepVidIndex_img2.jpg]

Group video frames into shots

[Figure by Wang & Suter]

Figure-ground



[Figure by Grauman & Darrell]

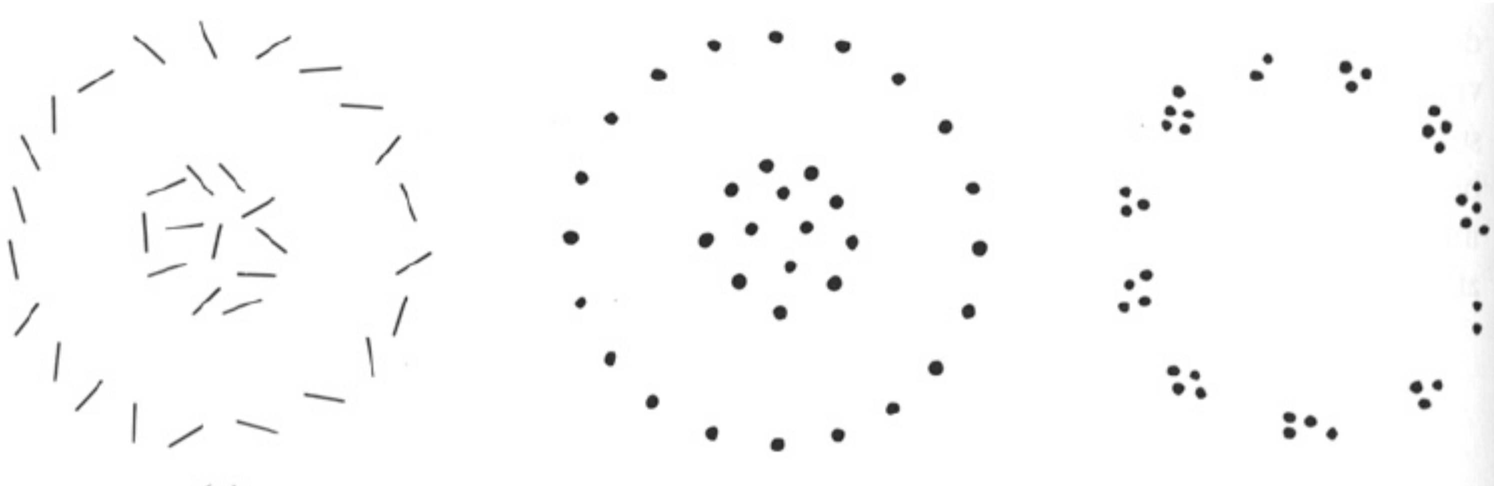
Object-level grouping

Slide credit: Kristen Grauman

Grouping in vision

- Goals:
  - Gather features that belong together
  - Obtain an intermediate representation that compactly describes key image (video) parts
- Top down vs. bottom up segmentation
  - Top down: pixels belong together because they are from the same object
  - Bottom up: pixels belong together because they look similar
- Hard to measure success
  - What is interesting depends on the application.

Slide credit: Kristen Grauman

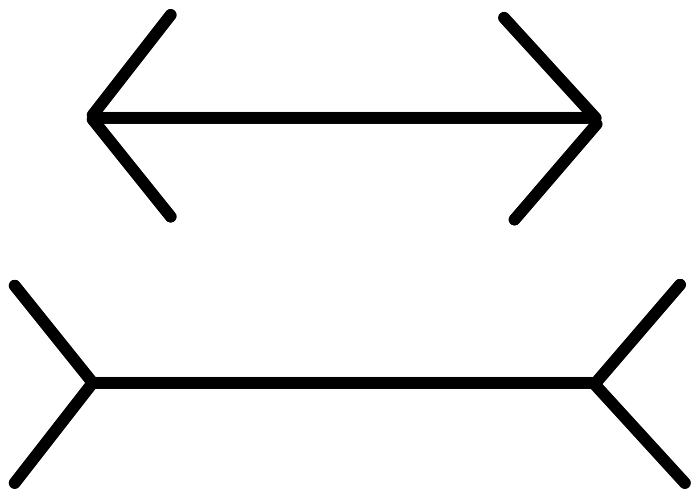


Slide credit: Kristen Grauman

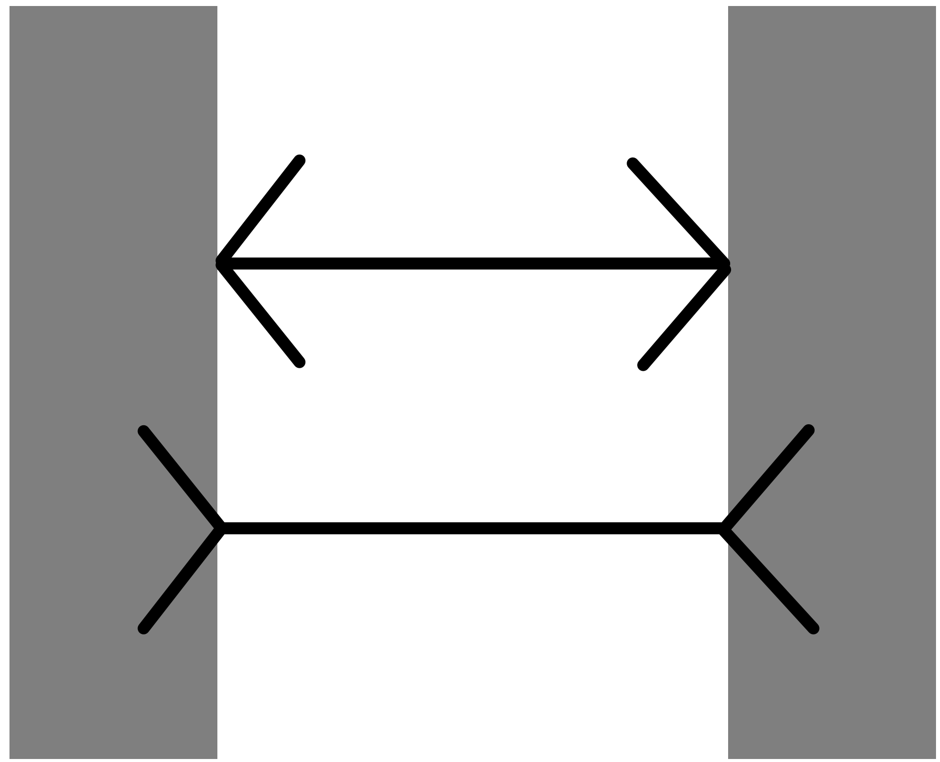
Gestalt

- A key feature of the human visual system is that context affects how things are perceived
- Gestalt: whole or group
  - Whole is something other than sum of its parts
  - Relationships among parts can yield new properties/features

Example: Muller-Lyer illusion

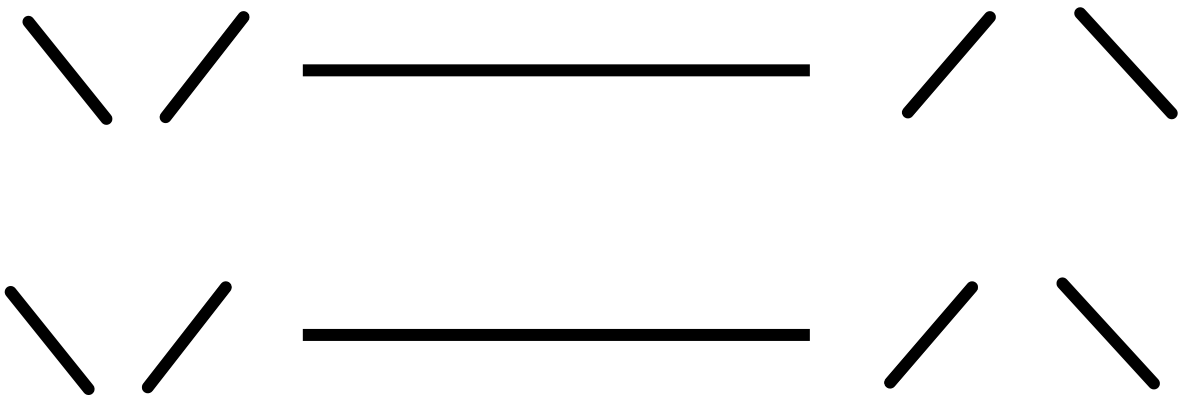


Example: Muller-Lyer illusion



Example: Muller-Lyer illusion

The effect *only* arises because we perceive each shape as something *other* than the sum of it's parts...



What things should be grouped?
What cues indicate groups?

Gestalt

- Gestalt: whole or group
  - Whole is something other 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)

Slide credit: Kristen Grauman

Gestalt

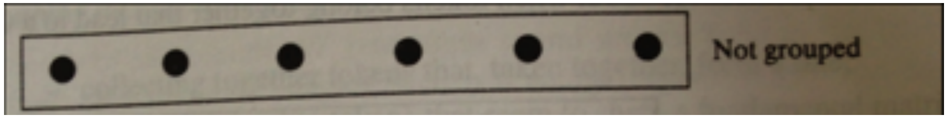
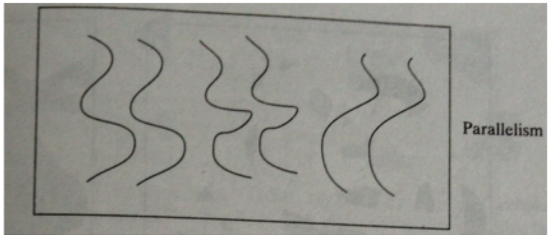


Figure 14.4 from Forsyth and Ponce

Slide credit: Devi Parikh

Gestalt



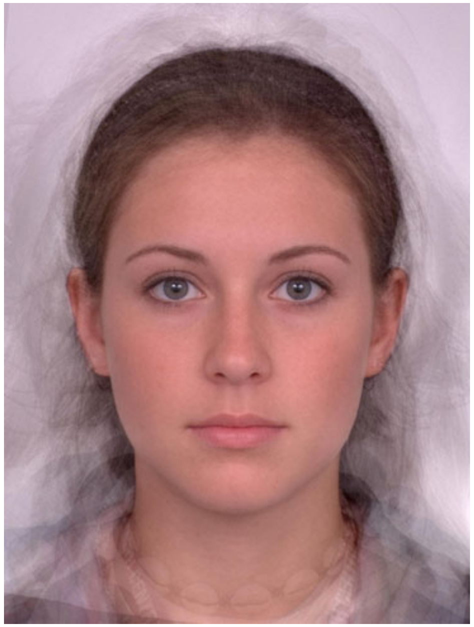
Slide credit: Devi Parikh

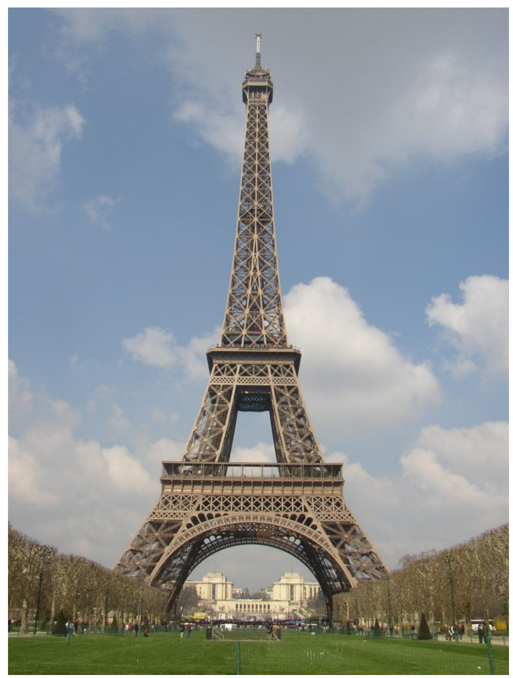
Similarity

Kristen Grauman

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

Symmetry





Slide credit: Kristen Grauman

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

Common fate

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

(coherent motion)

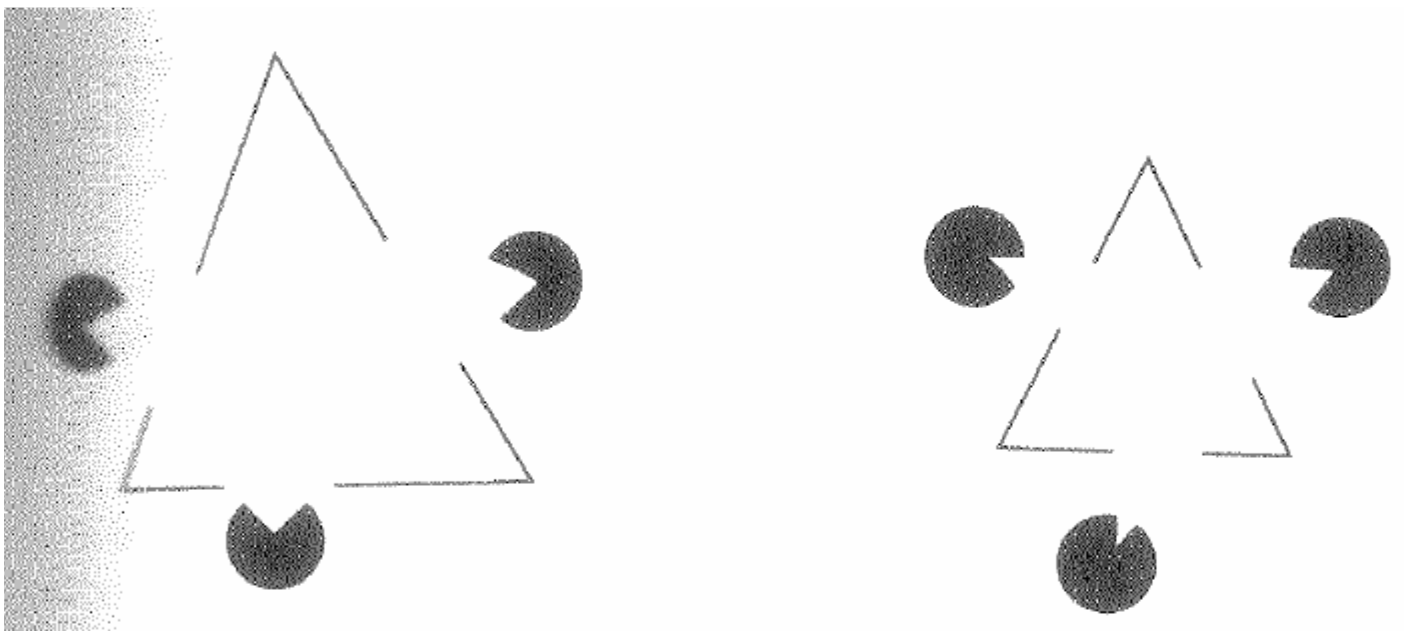
Slide credit: Kristen Grauman

Proximity

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

Slide credit: Kristen Grauman

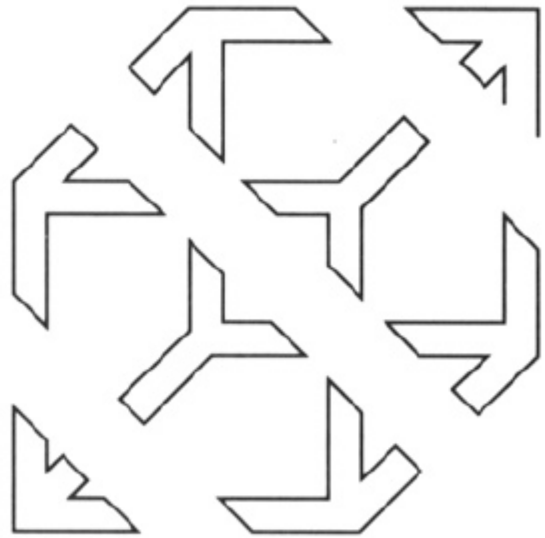
Illusory/subjective contours



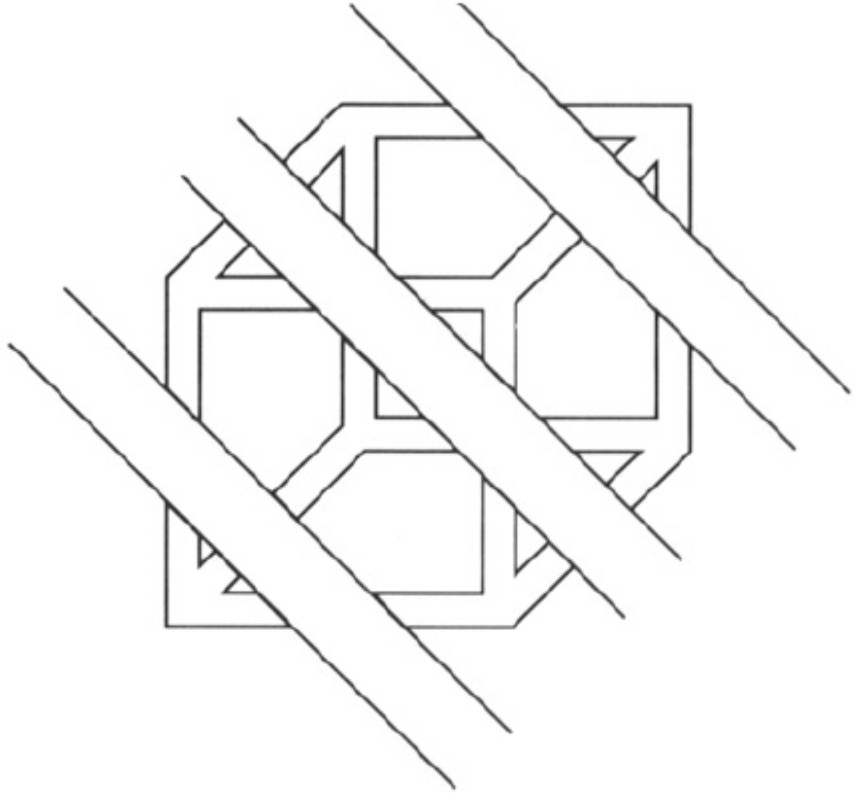
Interesting tendency to explain by occlusion

Slide credit: Kristen Grauman

In *Vision*, D. Marr, 1982

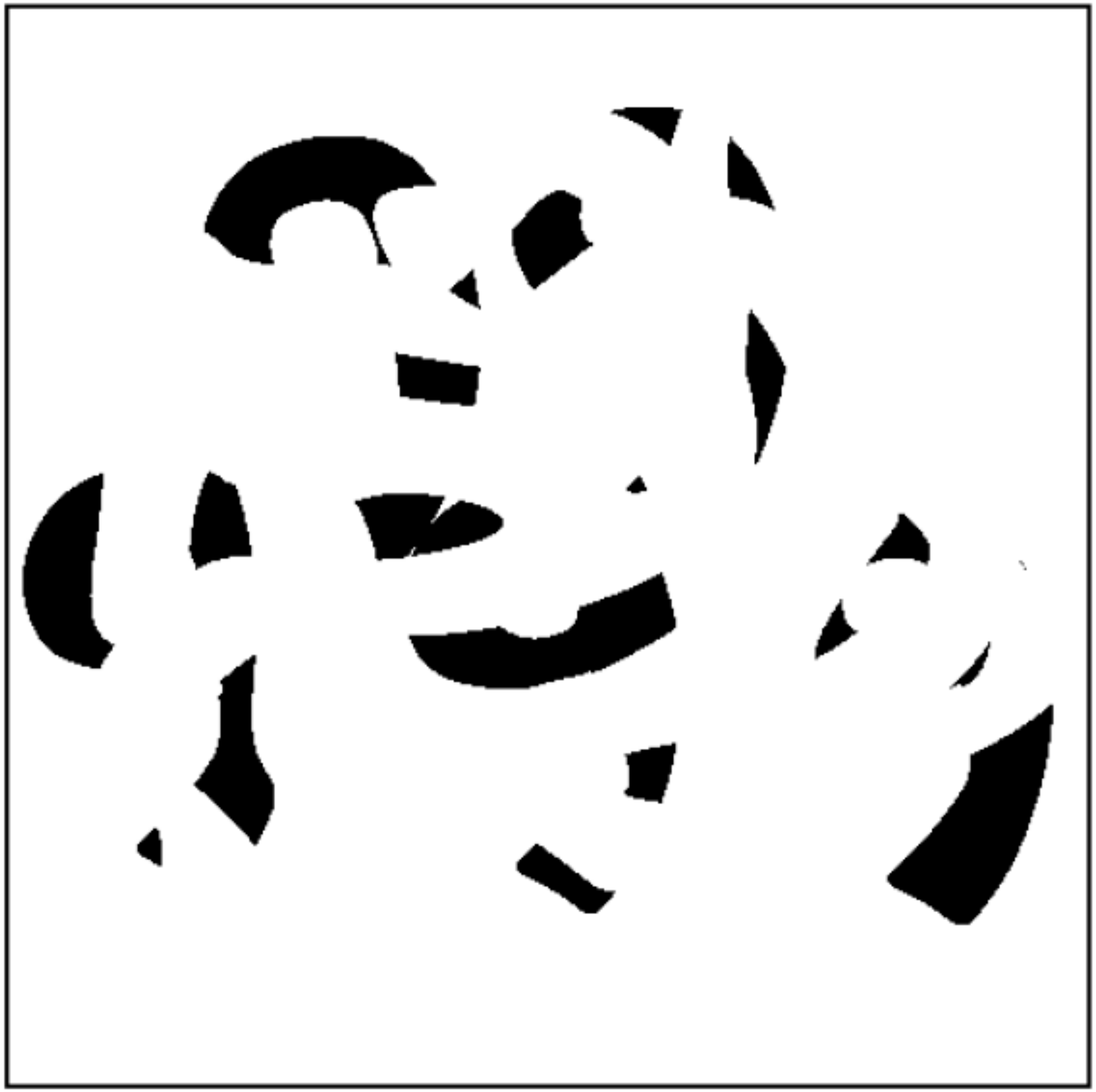


Slide credit: Kristen Grauman



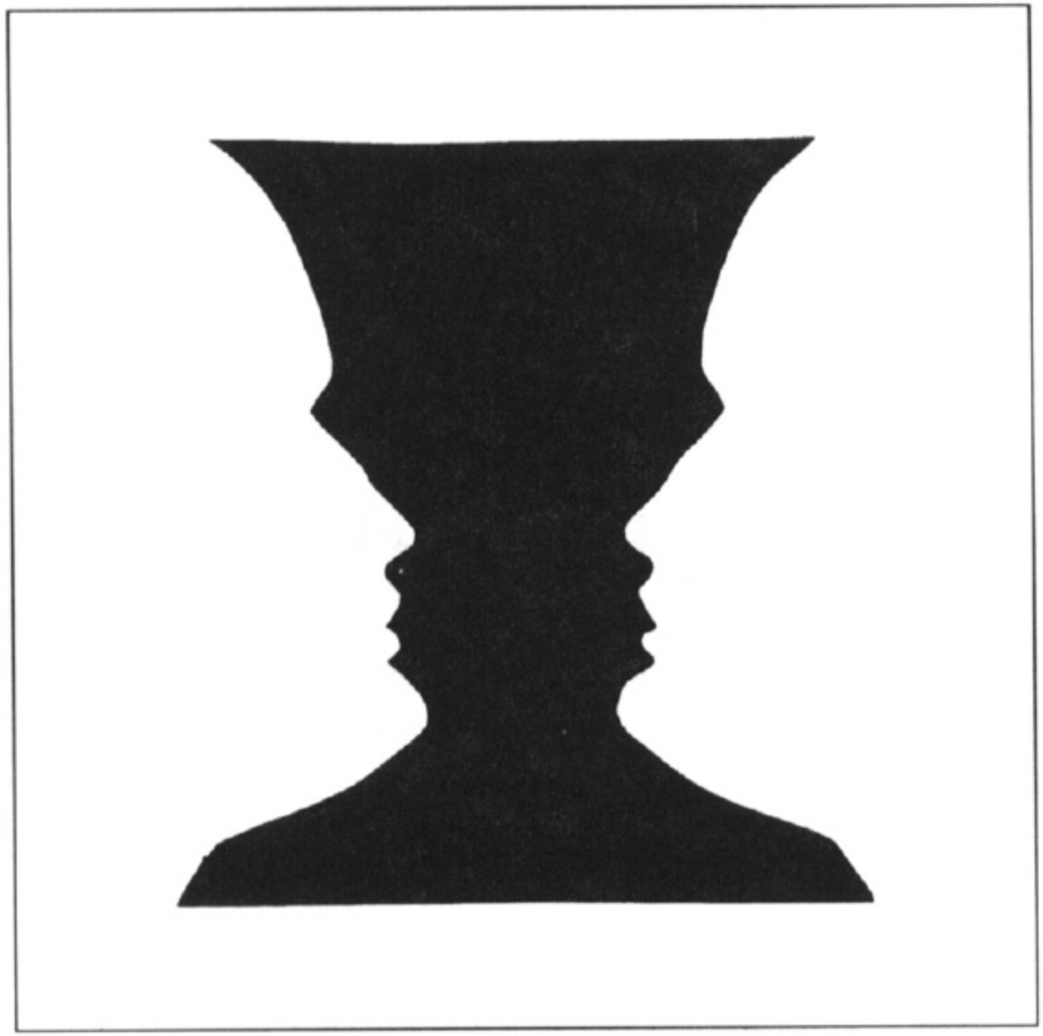
Continuity, explanation by occlusion

Slide credit: Kristen Grauman



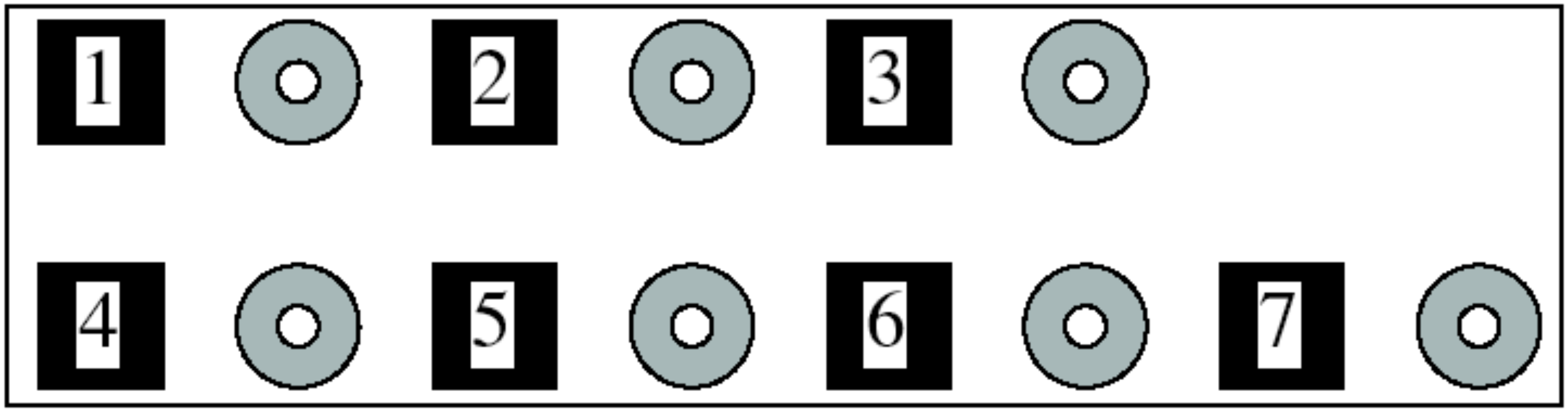
Slide credit: Kristen Grauman

Figure-ground



Slide credit: Kristen Grauman

Grouping phenomena in real life



Forsyth & Ponce, Figure 14.7

Slide credit: Kristen Grauman

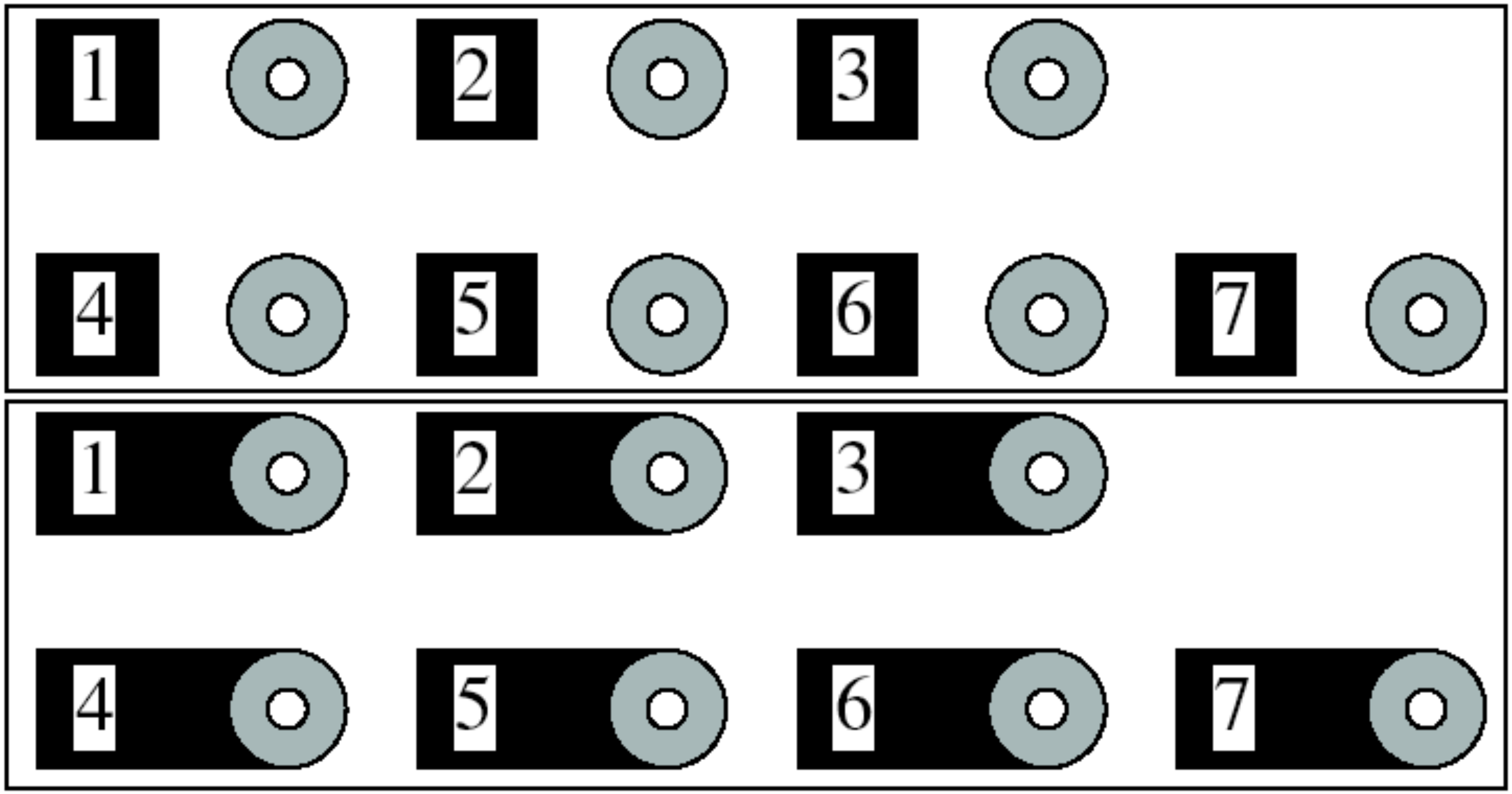
Grouping phenomena in real life

Figure 14.7 An example of grouping phenomena in real life. The buttons on an elevator in the computer science building at U.C. Berkeley used to be laid out as in the **top** figure. It was common to arrive at the wrong floor and discover that this was because you'd pressed the wrong button—the buttons are difficult to group unambiguously with the correct label, and it is easy to get the wrong grouping at a quick glance. A public-spirited individual filled in the gap between the numbers and the buttons, as in the **bottom** figure, and the confusion stopped because the proximity cue had been disambiguated.

Forsyth & Ponce, Figure 14.7

Slide credit: Kristen Grauman

Grouping phenomena in real life



Forsyth & Ponce, Figure 14.7

Slide credit: Kristen Grauman

Gestalt

- Gestalt: whole or group
  – Whole is other 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)

- Inspiring observations/explanations; challenge remains how to best map to algorithms.

Slide credit: Kristen Grauman

Grouping in Vision

Segmentation as Clustering

Mode finding & Mean-Shift

Graph-Based Algorithms

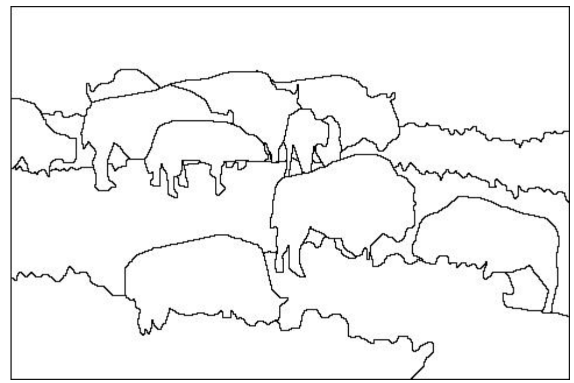
Segments as Primitives

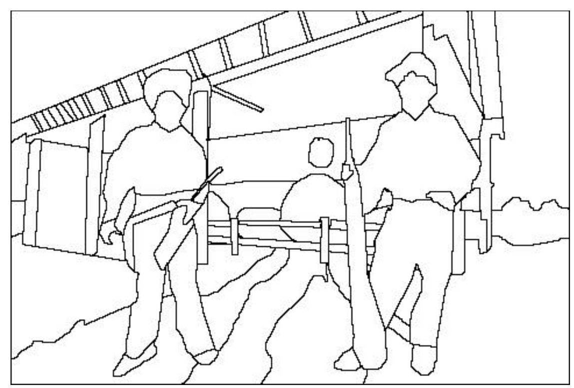
CNN-Based Approaches

The goals of segmentation

- Separate image into coherent “objects”

image | human segmentation

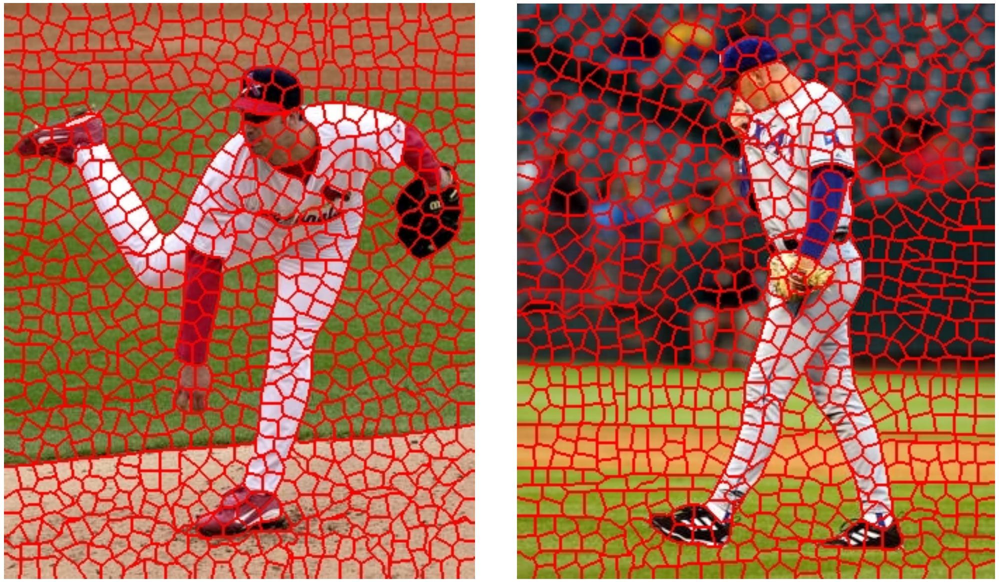
Source: Lana Lazebnik

The goals of segmentation

- Separate image into coherent “objects”

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

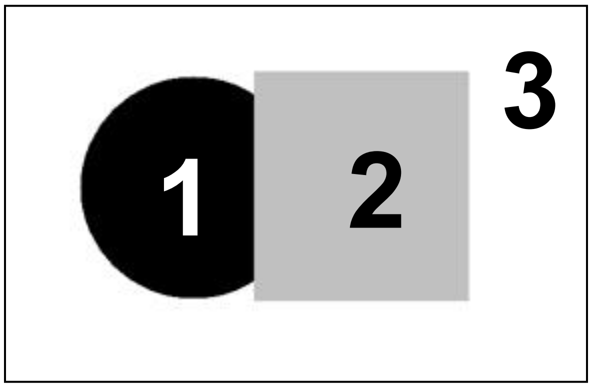
“superpixels”



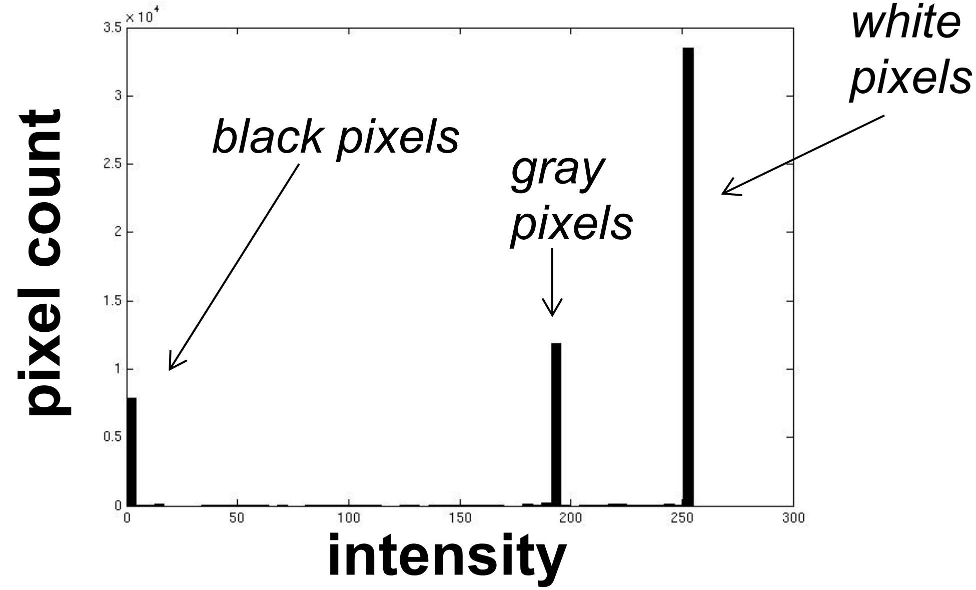
X. Ren and J. Malik. [Learning a classification model for segmentation.]() ICCV 2003.

Source: Lana Lazebnik

Image segmentation: toy example

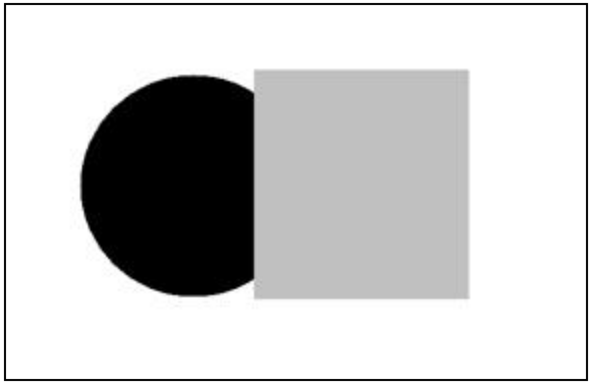


input image

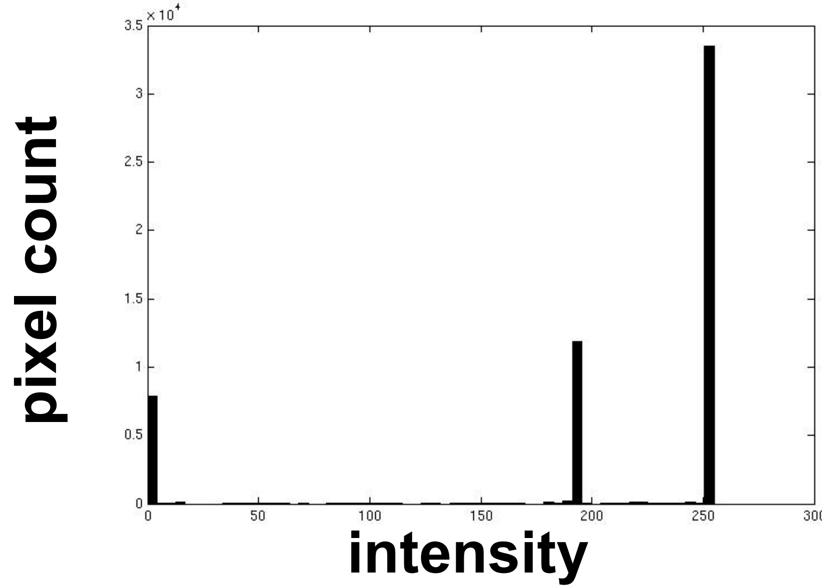


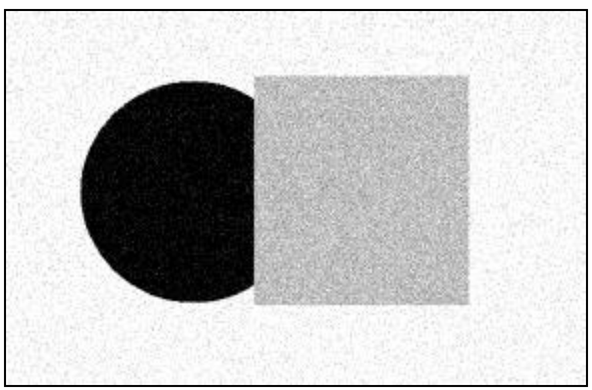
- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., *segment* the image based on the intensity feature.
- What if the image isn't quite so simple?

Kristen Grauman

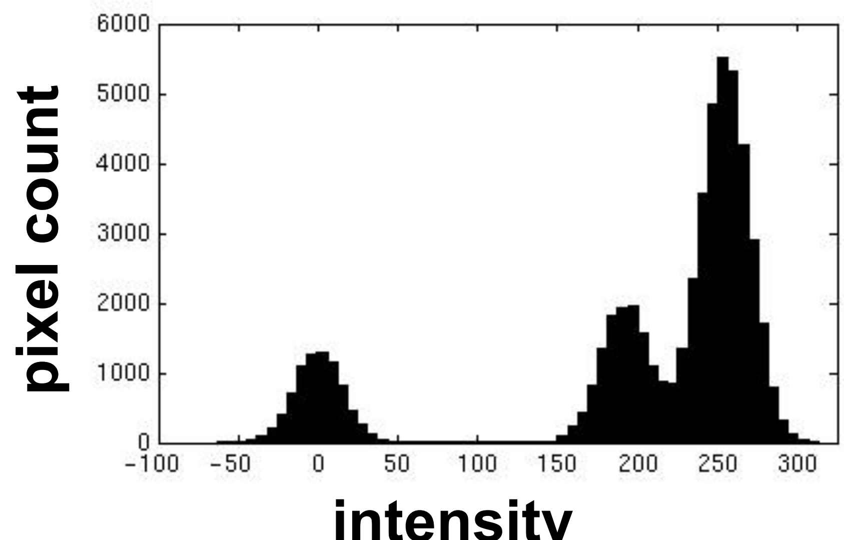


input image

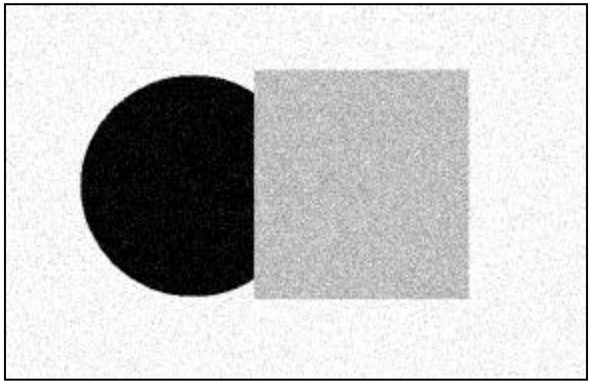




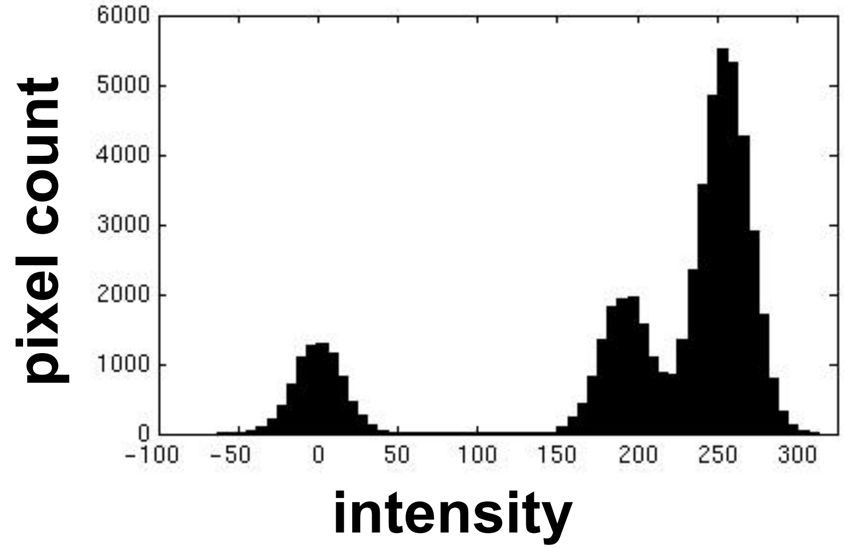
input image



Kristen Grauman

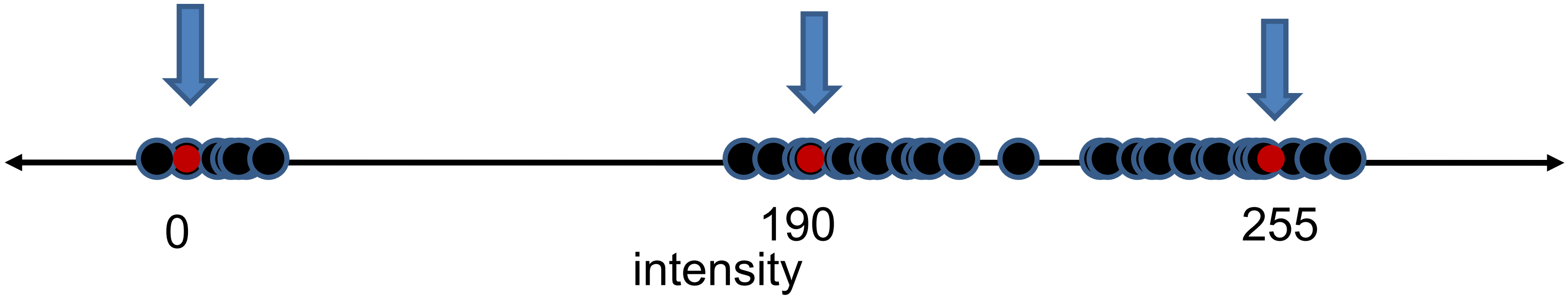


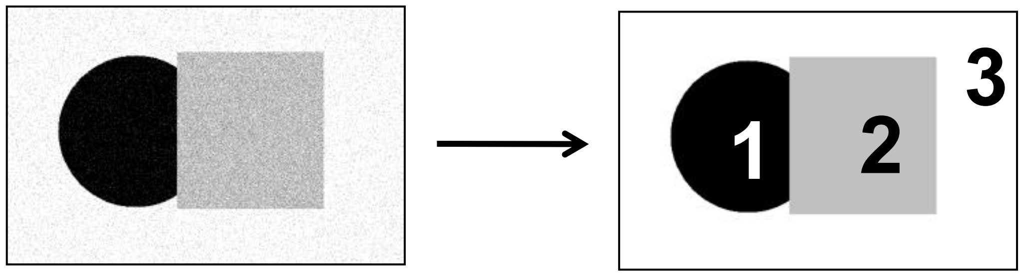
input image



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

Kristen Grauman



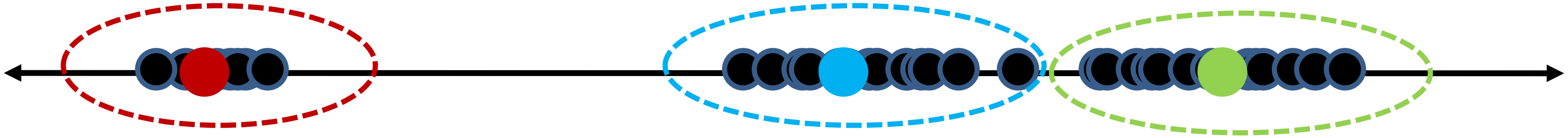


- Goal: choose three “centers” as the representative intensities, and label every pixel according to which of these centers it is nearest to.
- Best cluster centers are those that minimize SSD between all points and their nearest cluster center c_i:

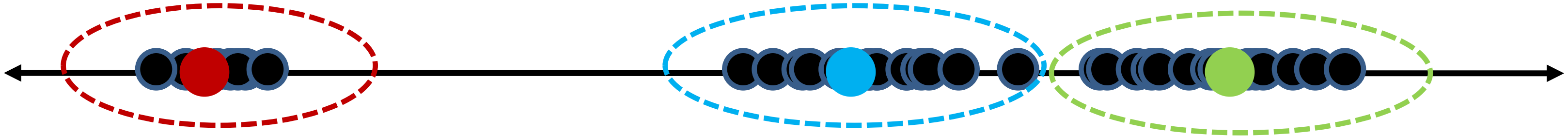
$$\sum_{\text{clusters } i} \; \sum_{\text{points p in cluster } i} ||p - c_i||^2$$

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.



Kristen Grauman

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:

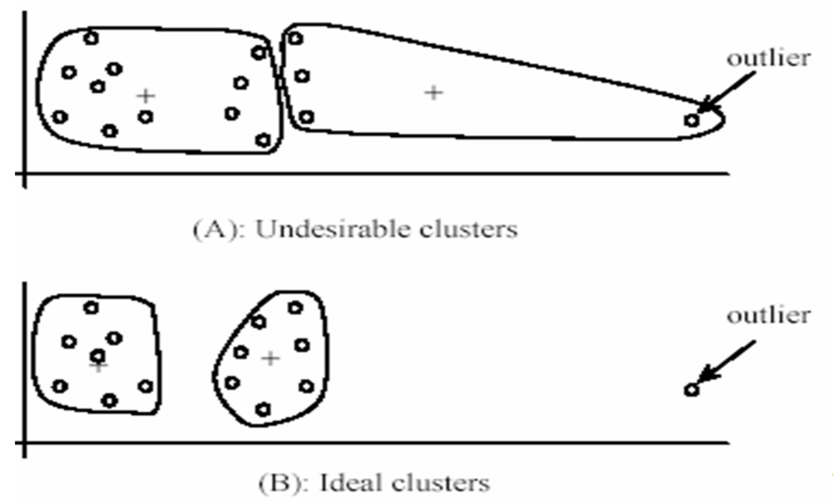
$$\sum_{\text{clusters } i} \sum_{\text{points p in cluster } i} \|p - c_i\|^2$$

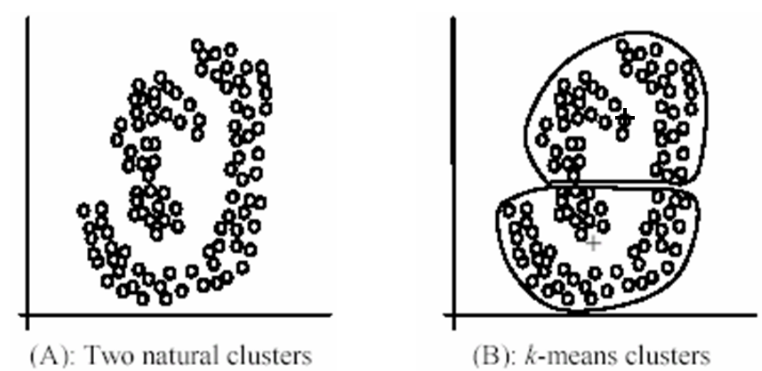
Source: Steve Seitz

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

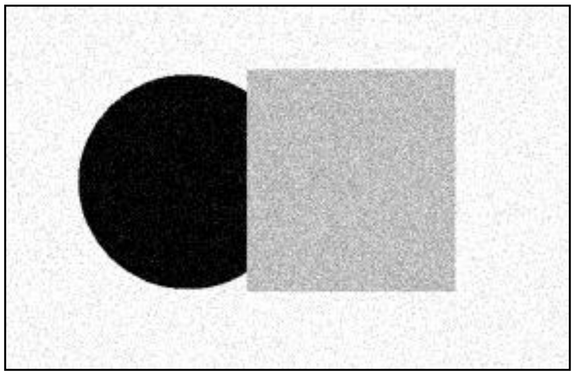




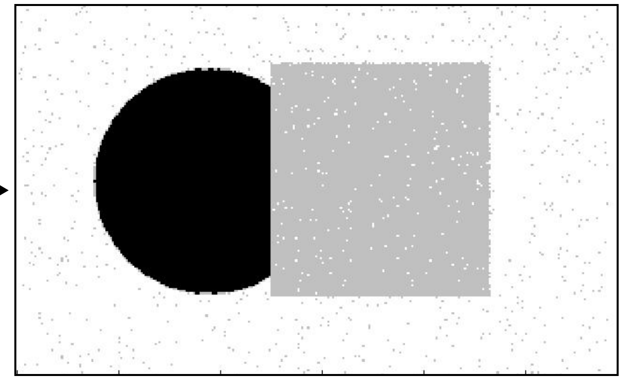
Slide credit: Kristen Grauman

An aside: Smoothing out cluster assignments

- Assigning a cluster label per pixel may yield outliers:



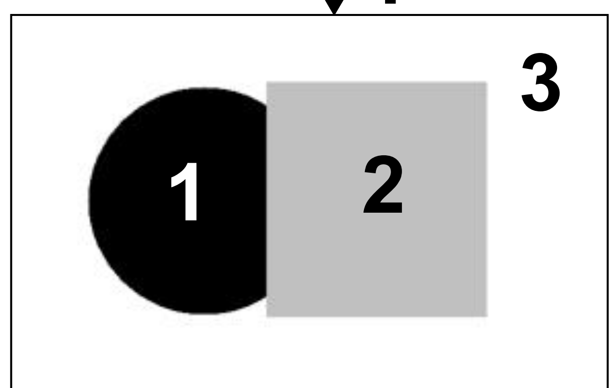
original



labeled by cluster center's intensity

?

- How to ensure they are spatially smooth?



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** similarity

Feature space: intensity value (1-d)

Slide credit: Kristen Grauman

K=2

K=3

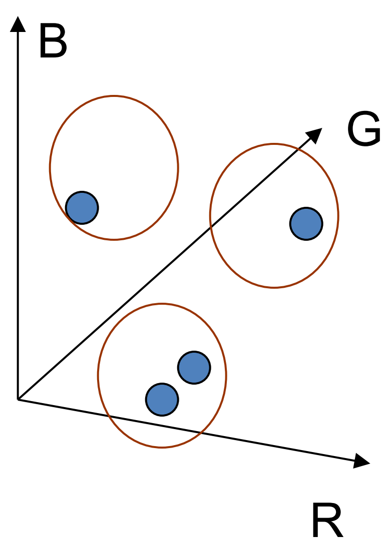
quantization of the feature space; segmentation label map

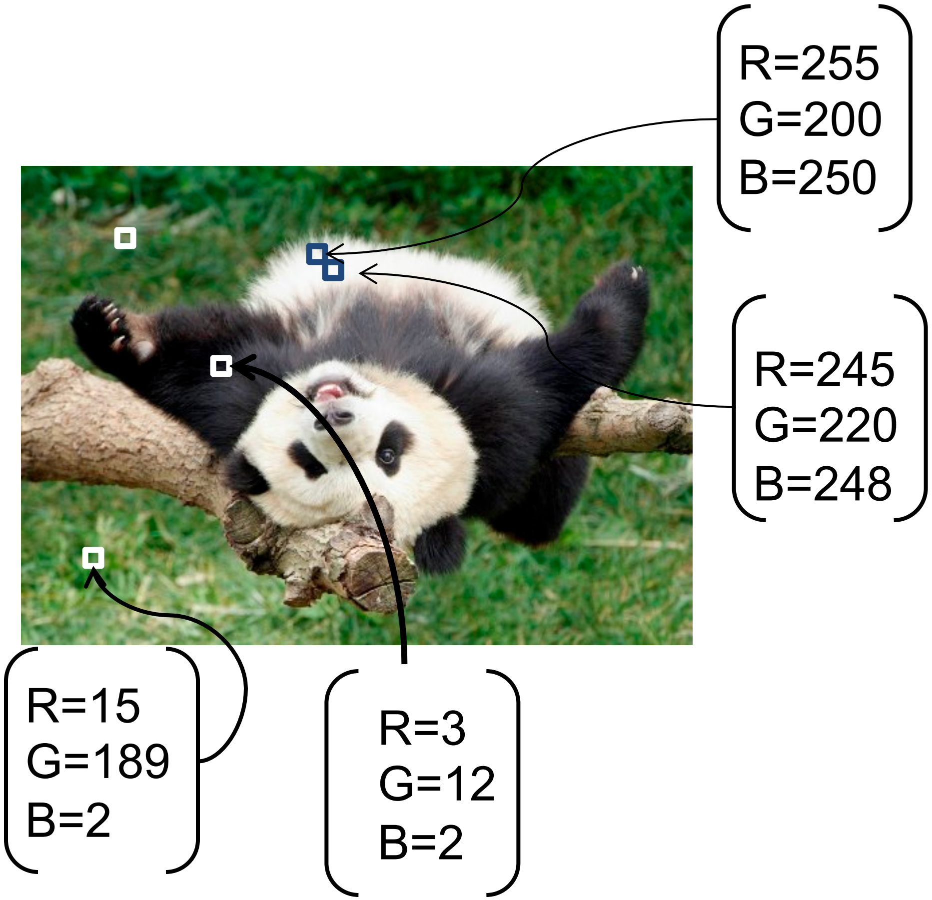
Slide credit: Kristen Grauman

Segmentation as clustering

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

Grouping pixels based on **color** similarity





Feature space: color value (3-d)

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** similarity

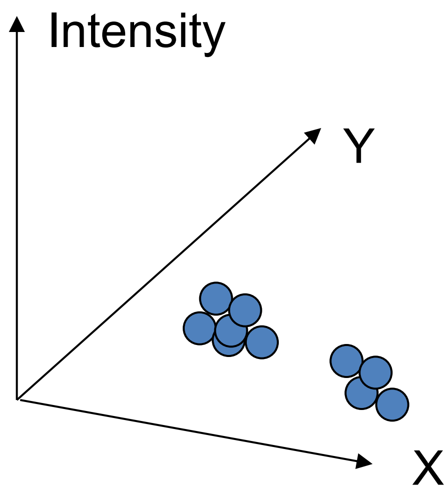
Clusters based on intensity similarity don't have to be spatially coherent.

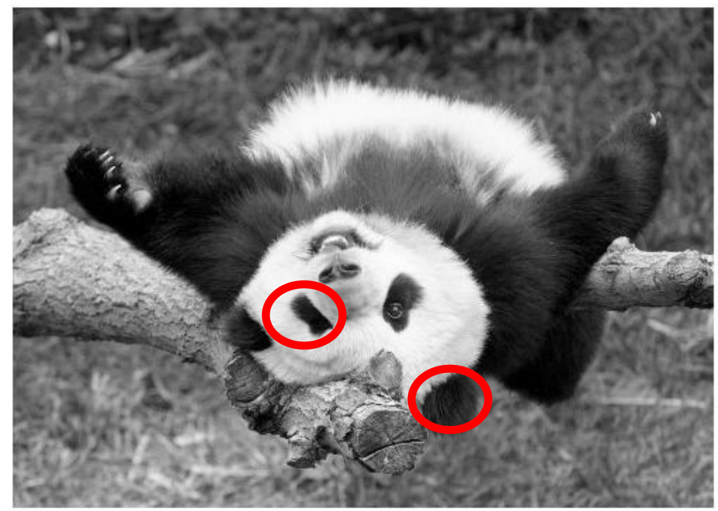
Kristen Grauman

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





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

Kristen Grauman

Segmentation as clustering

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

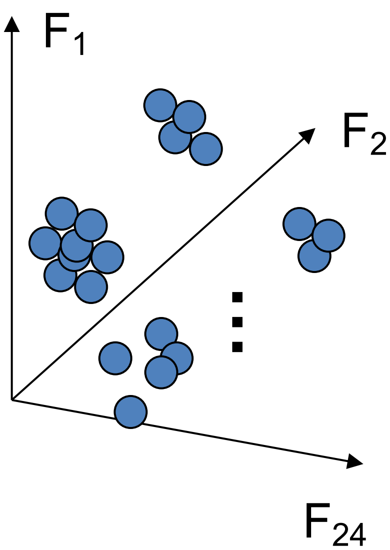


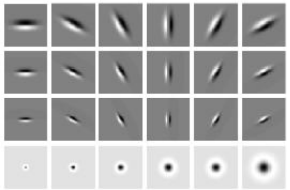
Slide credit: Kristen Grauman

Segmentation as clustering

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

Grouping pixels based on **texture** similarity





Filter bank of 24 filters

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

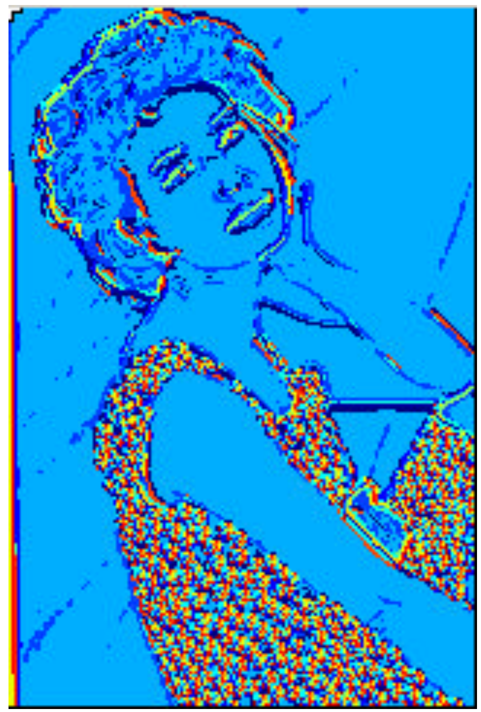
Slide credit: Kristen Grauman

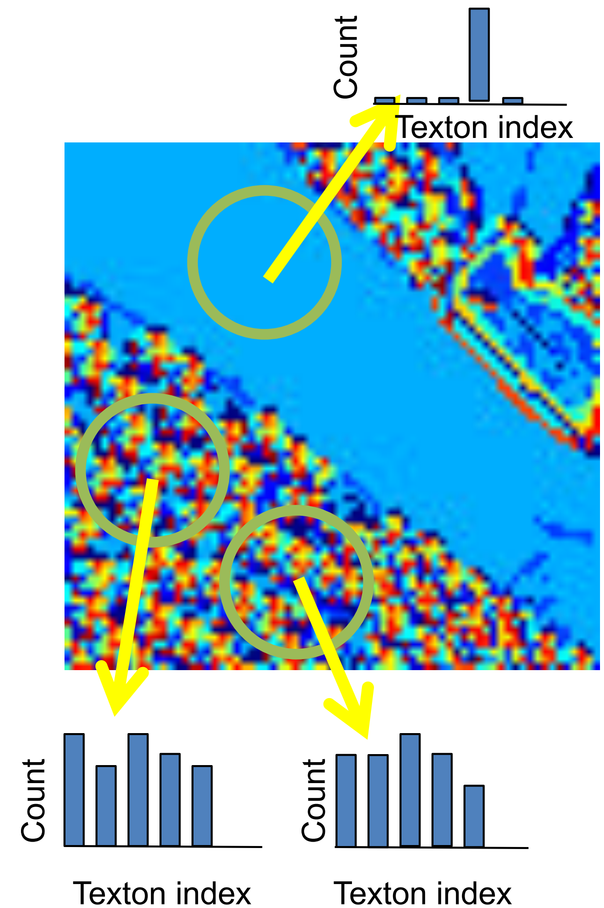
Segmentation with texture features

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

Image

Texton map

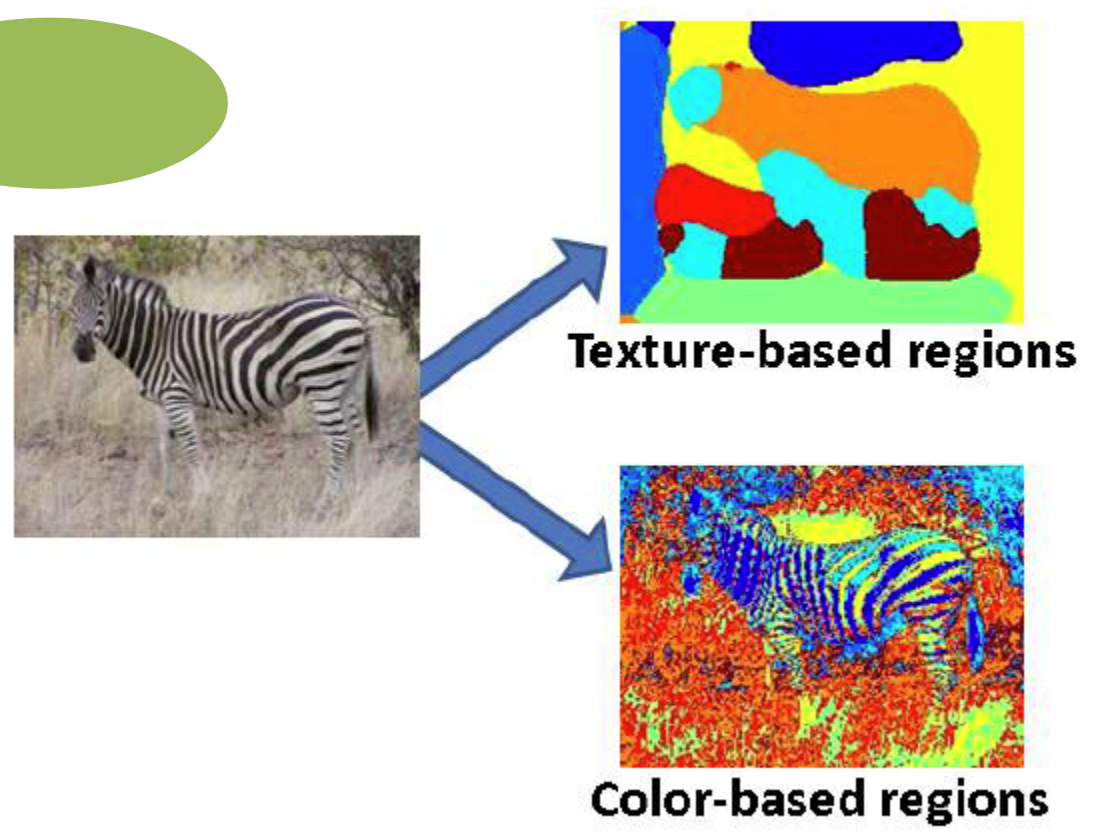




Malik, Belongie, Leung and Shi. IJCV 2001.

Adapted from Lana Lazebnik

Image segmentation example



Kristen Grauman

Pixel properties vs. neighborhood properties

These look very similar in terms of their color distributions (histograms).

How would their *texture* distributions compare?

Kristen Grauman

Grouping in Vision

Segmentation as Clustering

$\boxed{\text{Mode finding \& Mean-Shift}}$

Graph-Based Algorithms

Segments as Primitives

CNN-Based Approaches

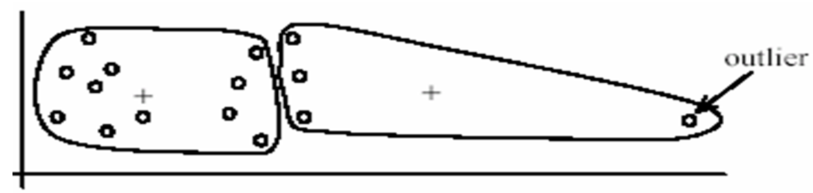
K-means: pros and cons

Pros

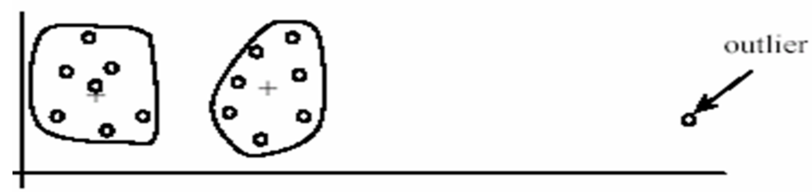
- Simple, fast to compute
- Converges to local minimum of within-cluster squared error

Cons/issues

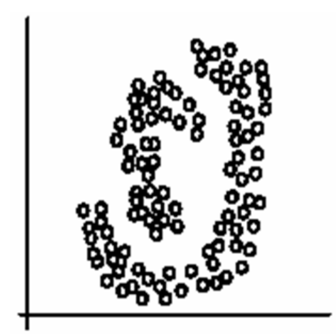
- Setting k?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters
- Assuming means can be computed



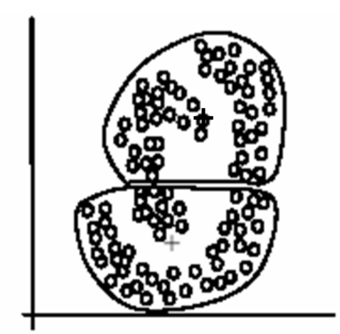
(A): Undesirable clusters



(B): Ideal clusters



(A): Two natural clusters



(B): *k*-means clusters

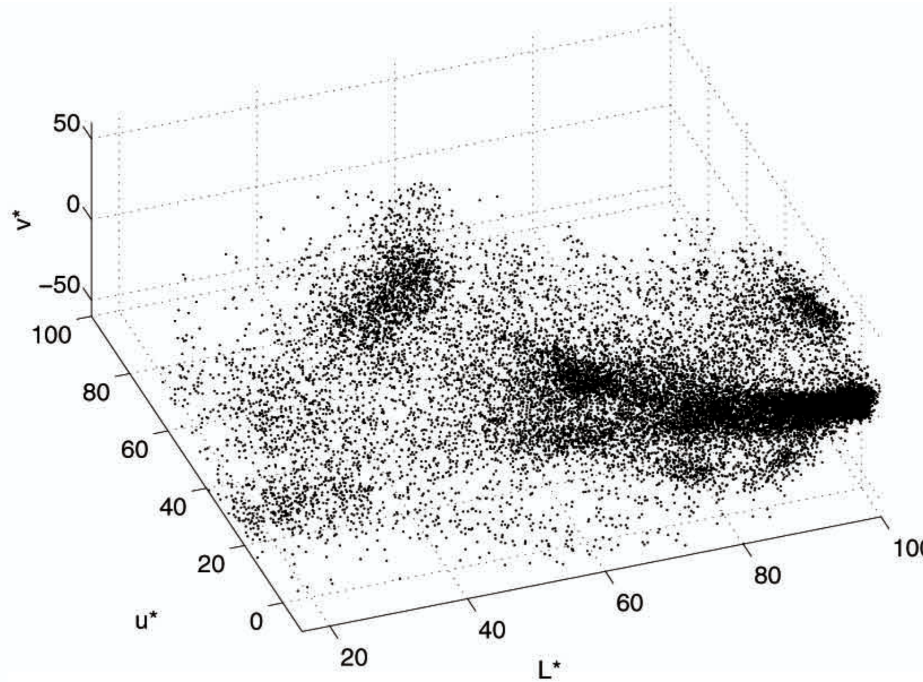
Slide credit: Kristen Grauman

Mean shift algorithm

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

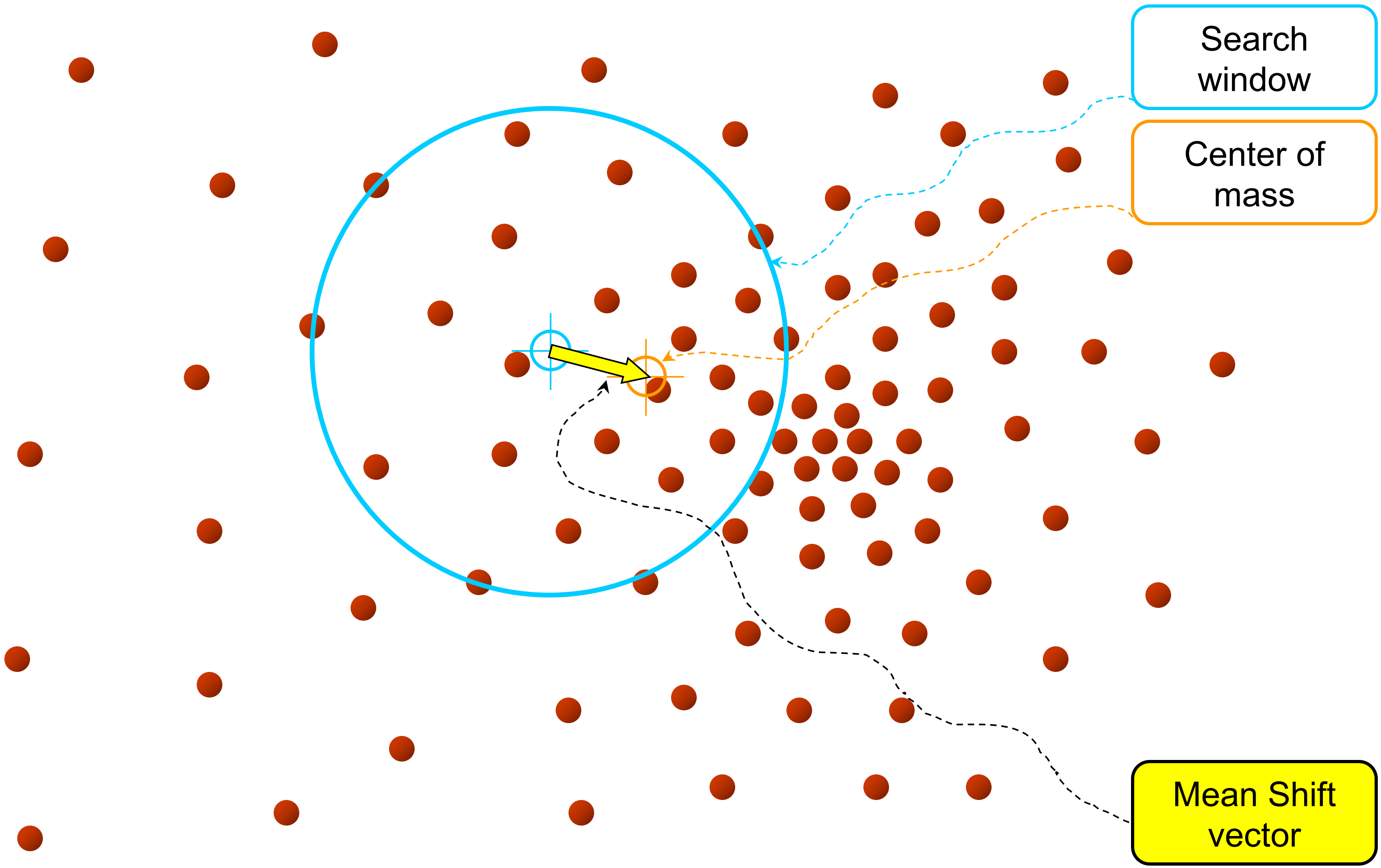
image

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



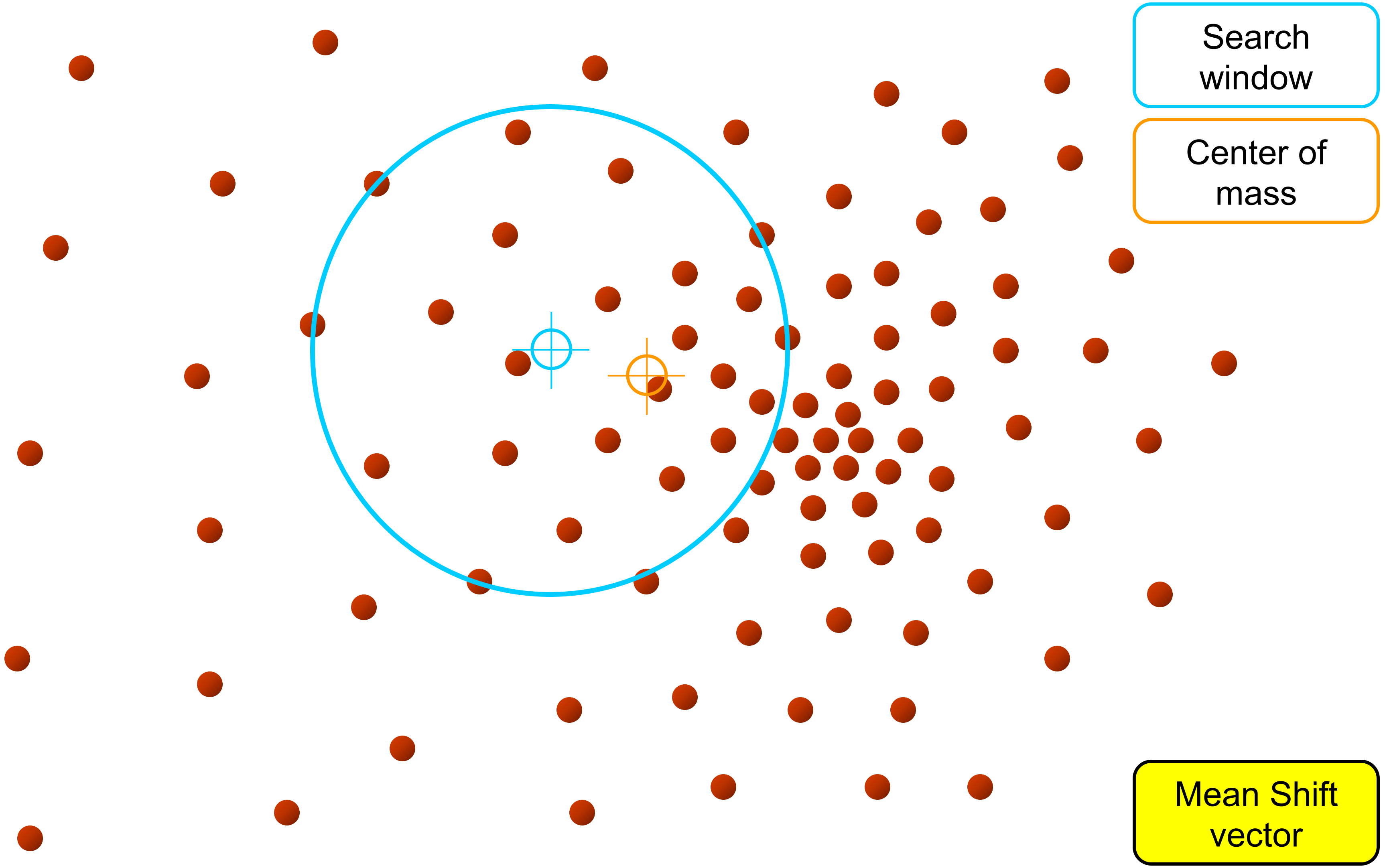
Slide credit: Kristen Grauman

Mean shift



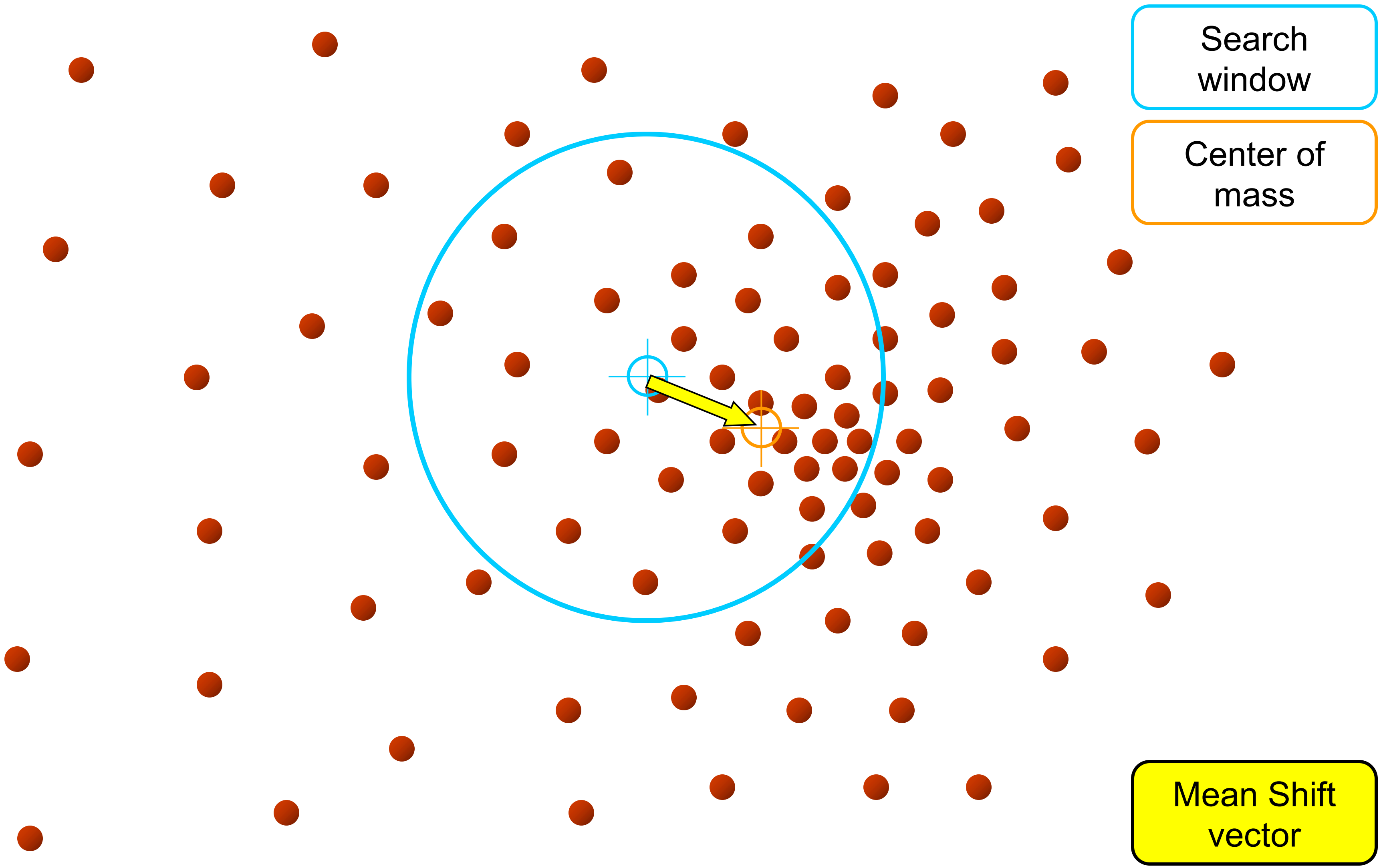
Slide by Y. Ukrainitz & B. Sarel

Mean shift



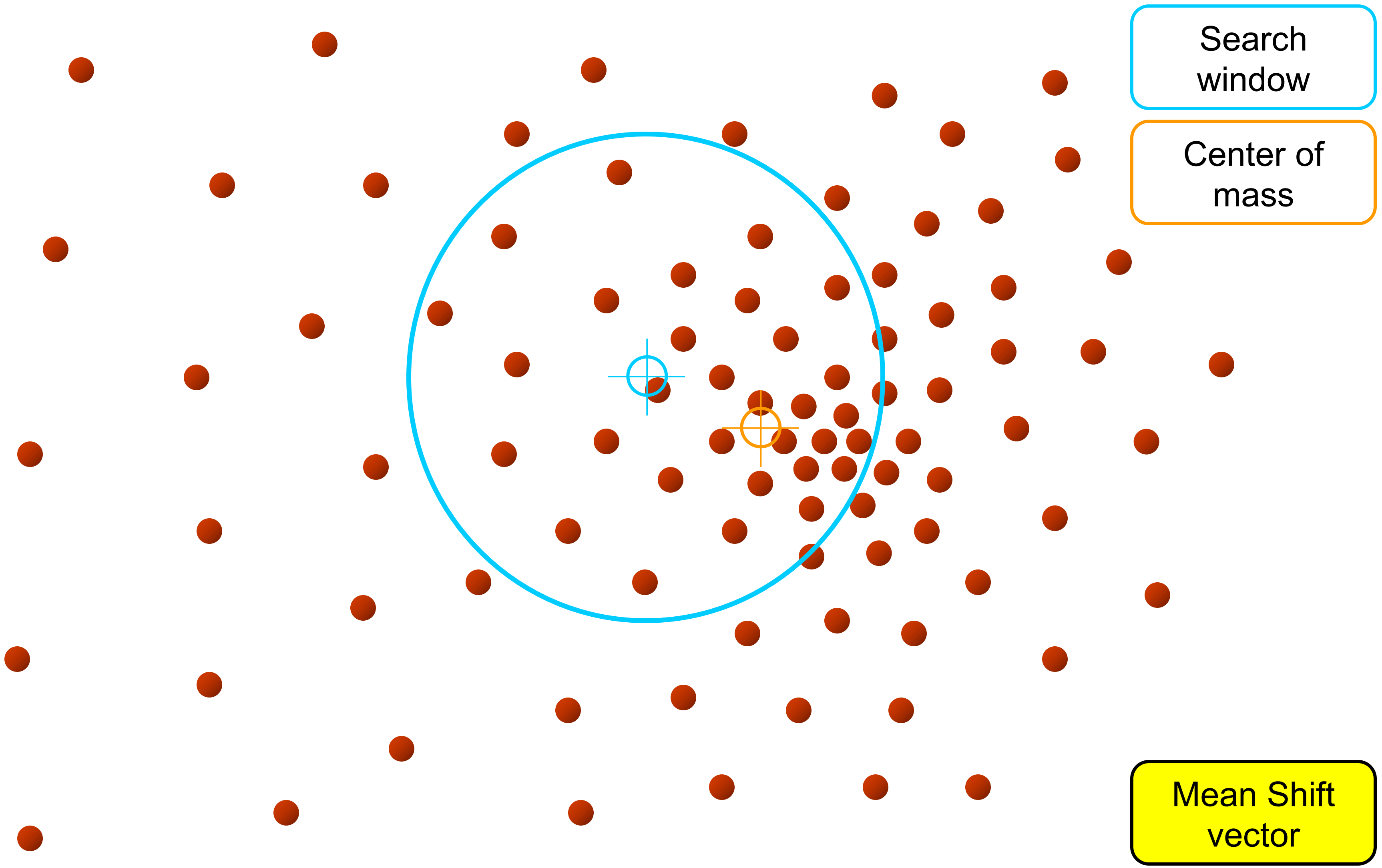
Slide by Y. Ukrainitz & B. Sarel

Mean shift



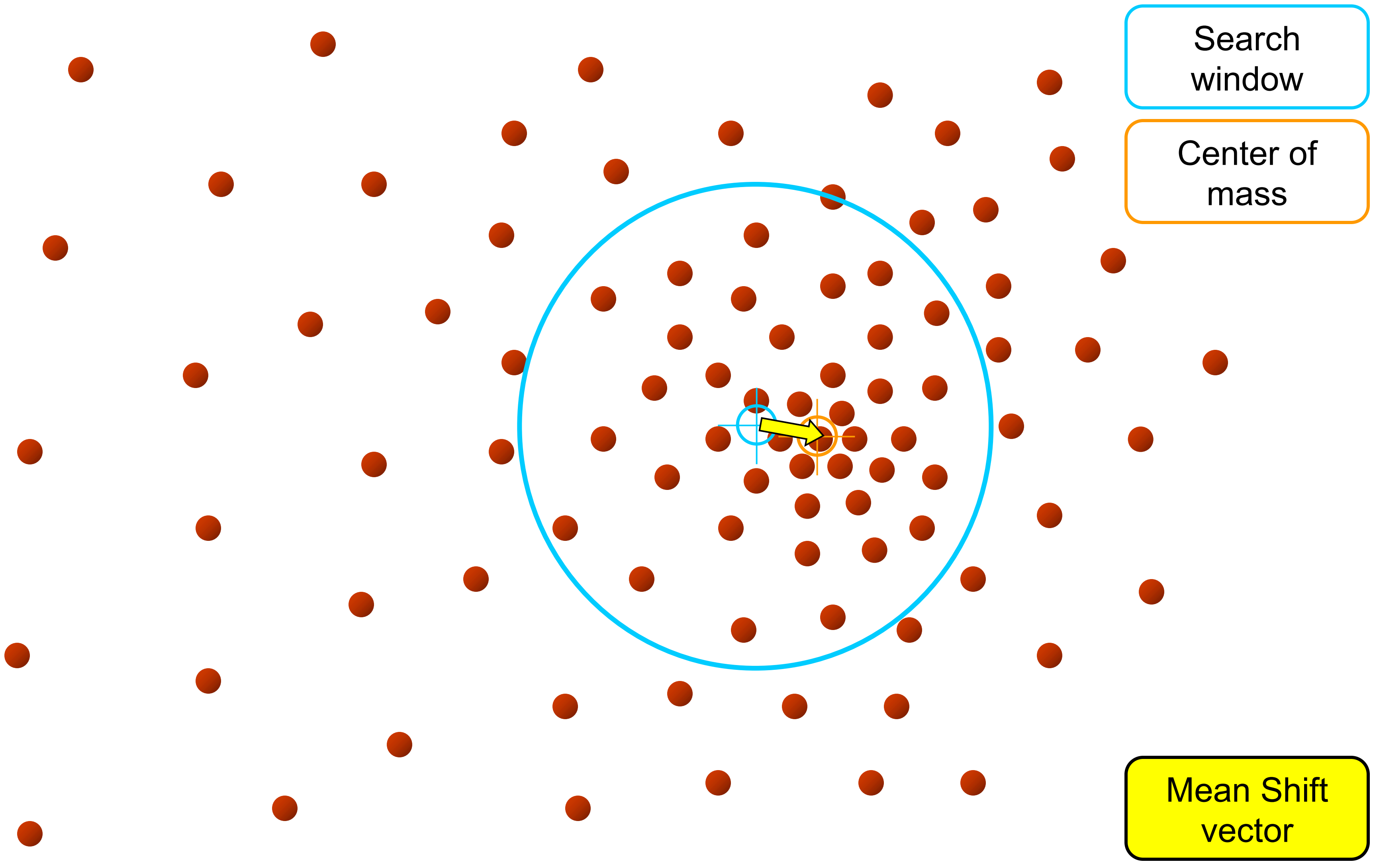
Slide by Y. Ukrainitz & B. Sarel

Mean shift



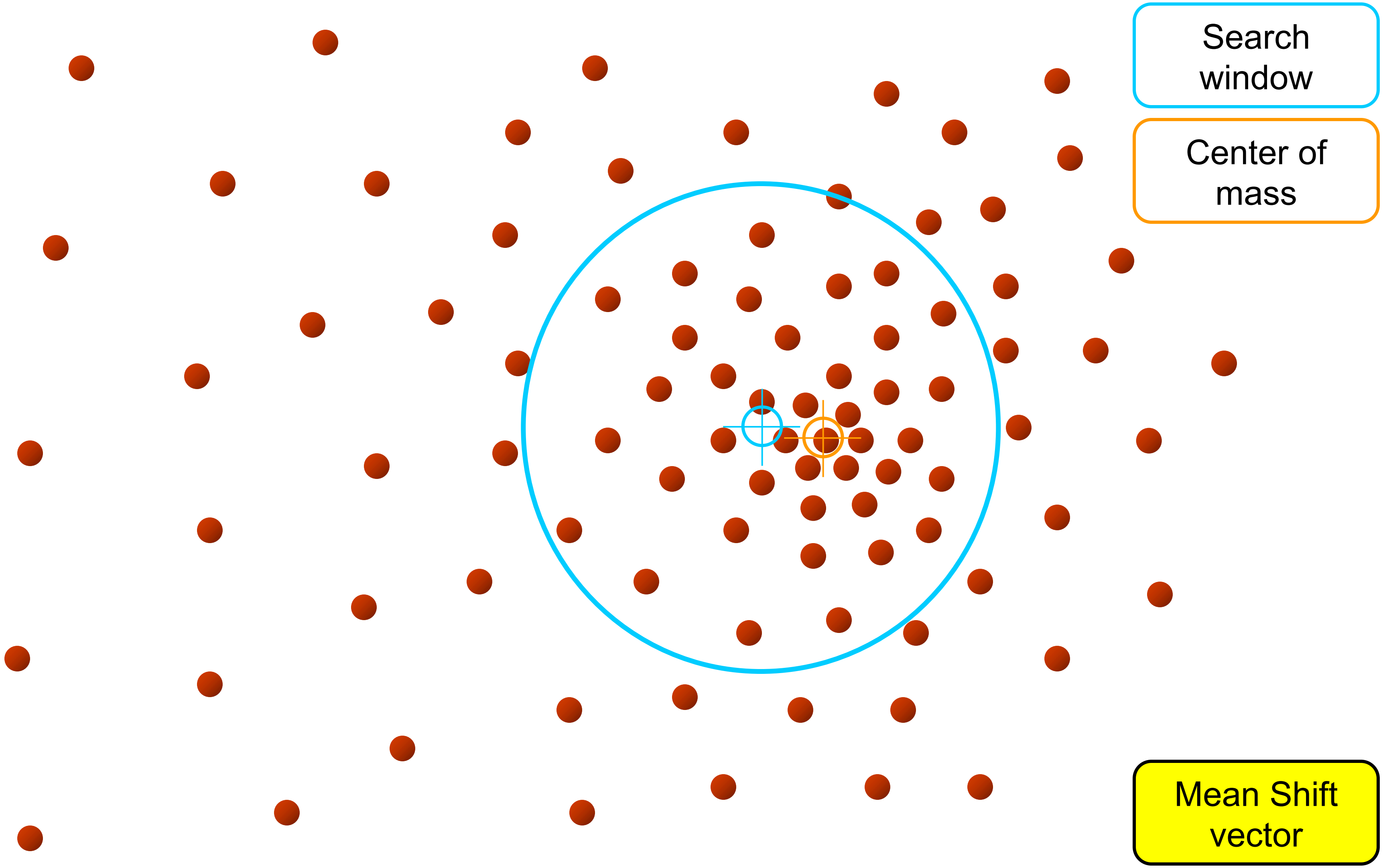
Slide by Y. Ukrainitz & B. Sarel

Mean shift



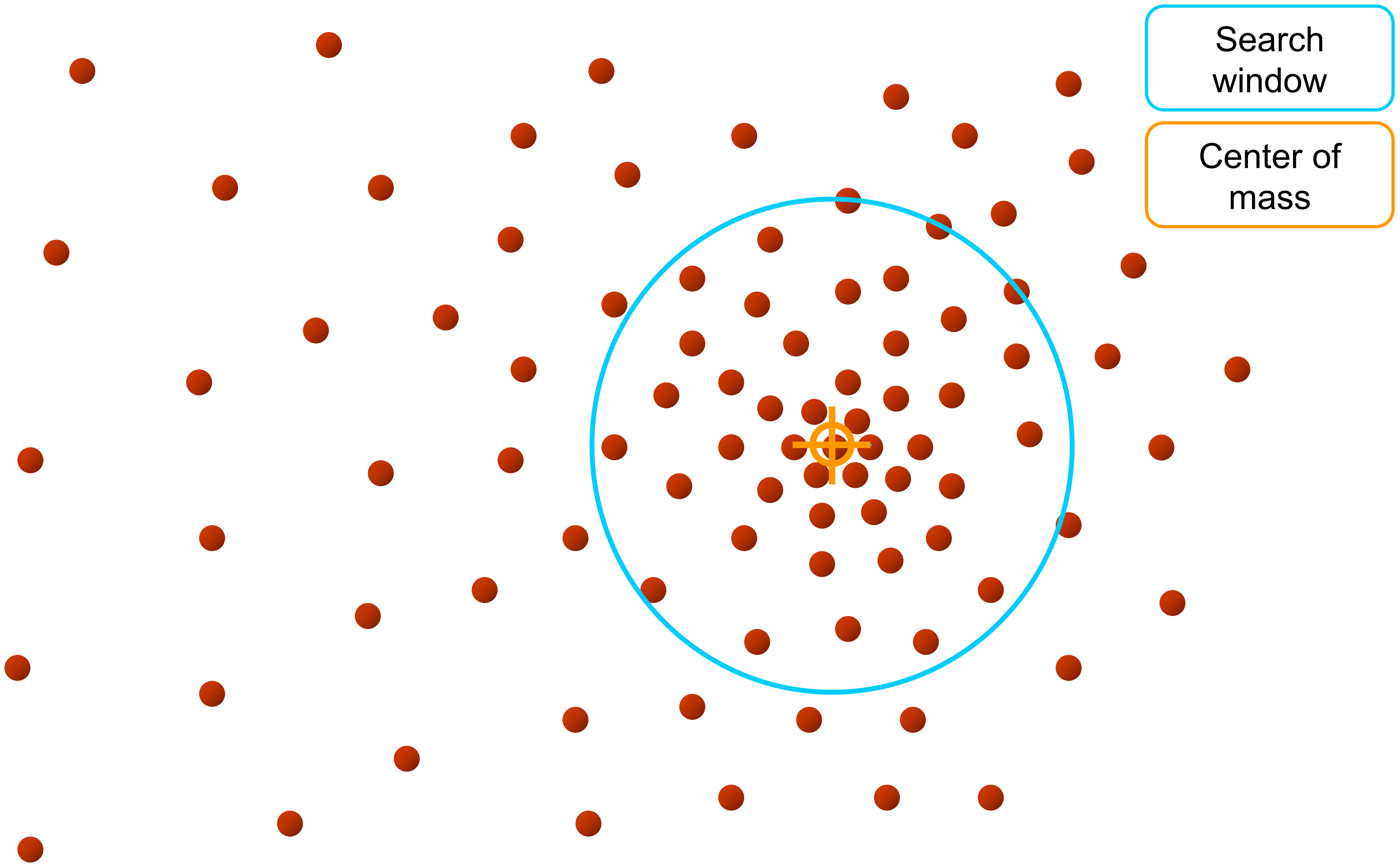
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

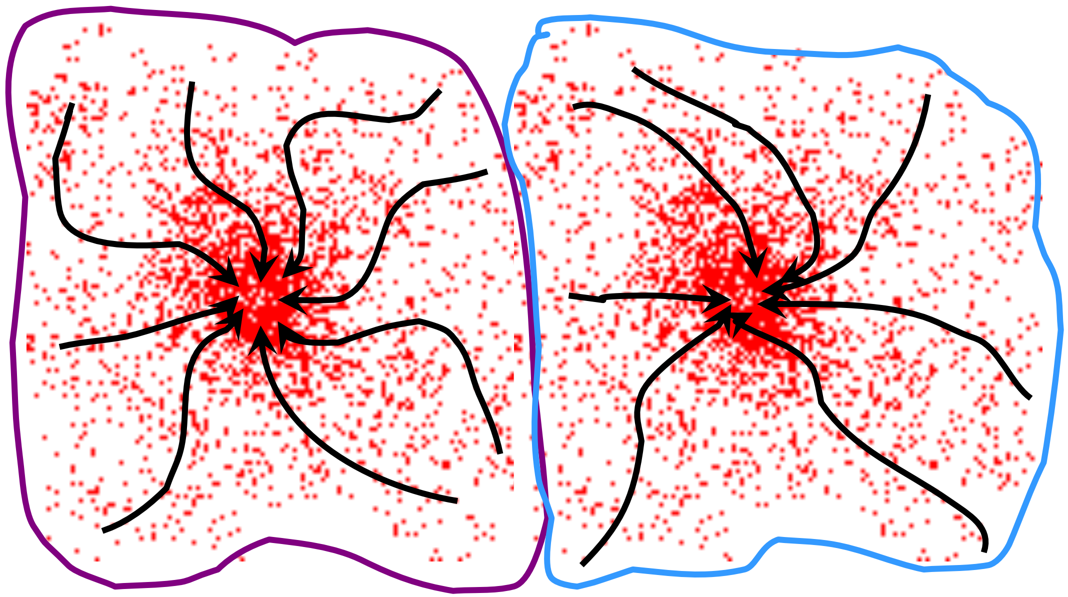
Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift clustering

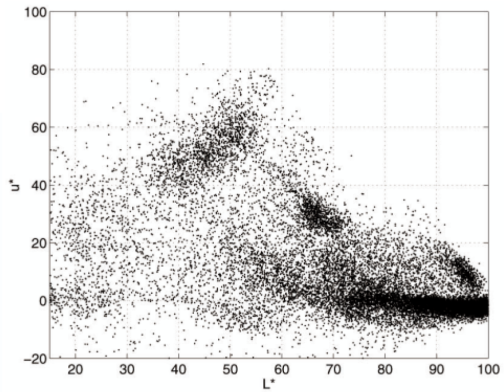
- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode



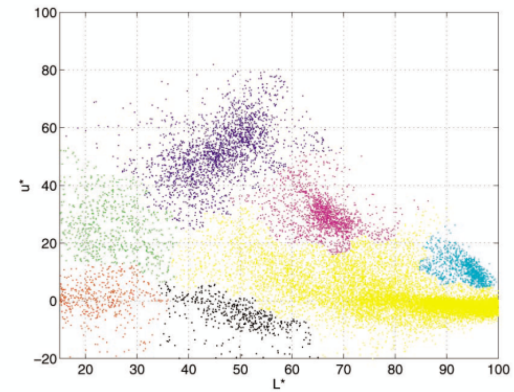
Slide by Y. Ukrainitz & B. Sarel

Mean shift clustering/segmentation

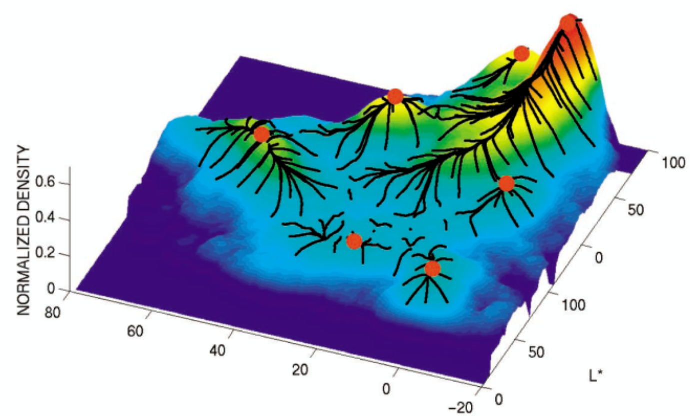
- Find features (color, gradients, texture, etc)
- Initialize windows at individual feature points
- Perform mean shift for each window until convergence
- Merge windows that end up near the same “peak” or mode



(a)



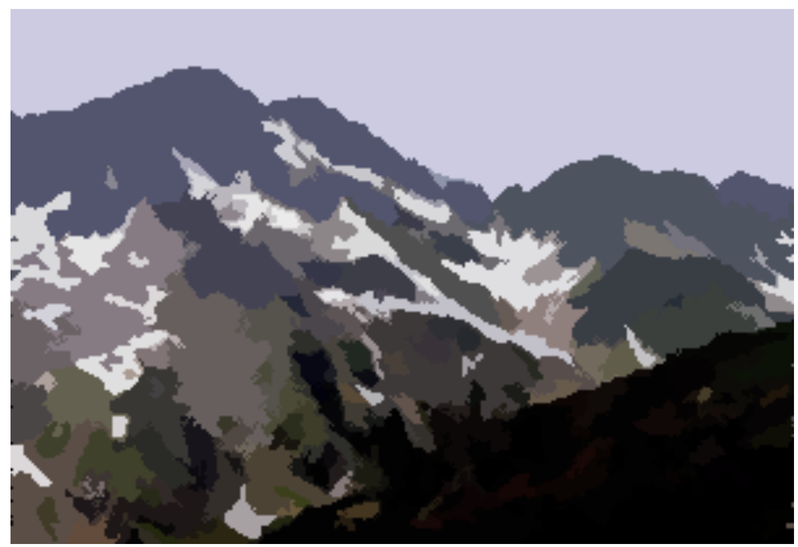
(b)



Slide credit: Kristen Grauman

Mean shift segmentation results



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

Slide credit: Kristen Grauman

Mean shift segmentation results

Slide credit: Kristen Grauman

Mean shift

- Pros:
  - Does not assume shape on clusters
  - One parameter choice (window size)
  - Generic technique
  - Find multiple modes
- Cons:
  - Selection of window size
  - Does not scale well with dimension of feature space

Kristen Grauman

Grouping in Vision

Segmentation as Clustering

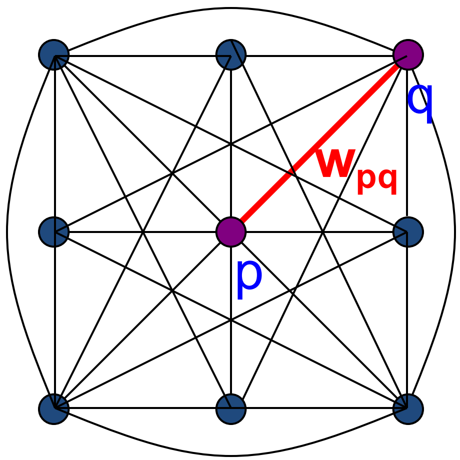
Mode finding & Mean-Shift

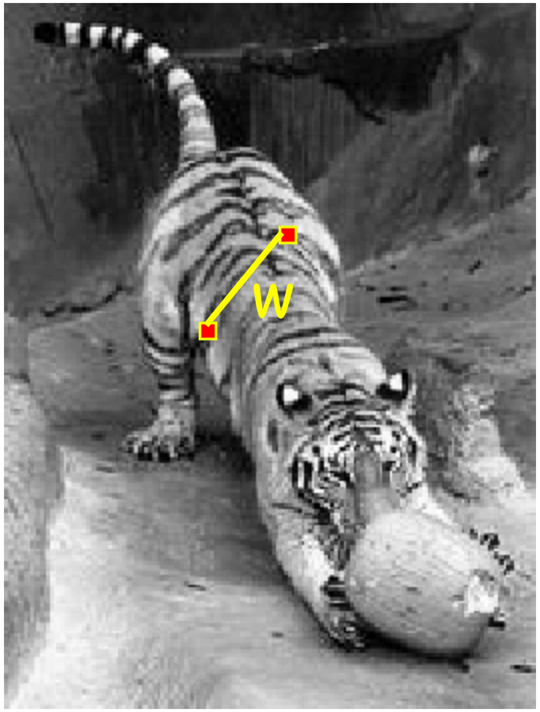
Graph-Based Algorithms

Segments as Primitives

CNN-Based Approaches

Images as graphs





- *Fully-connected* graph
  - node (vertex) for every pixel
  - link between *every* pair of pixels, **p**,**q**
  - affinity weight $\mathbf{w_{pq}}$ for each link (edge)
    - $\mathbf{w_{pq}}$ measures *similarity*
      - similarity is *inversely proportional* to difference (color+position…)

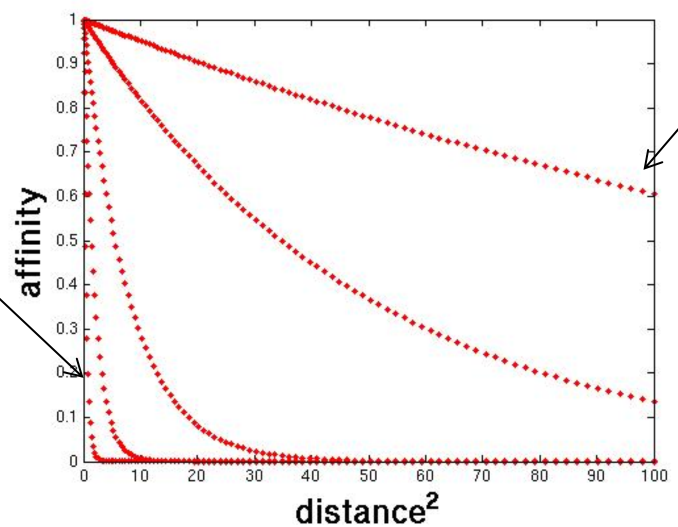
Source: Steve Seitz

Measuring affinity

- One possibility:

$$aff(x,y) = \exp\left\{-\left(\frac{1}{2\sigma_d^2}\right)\left(\|x-y\|^2\right)\right\}$$

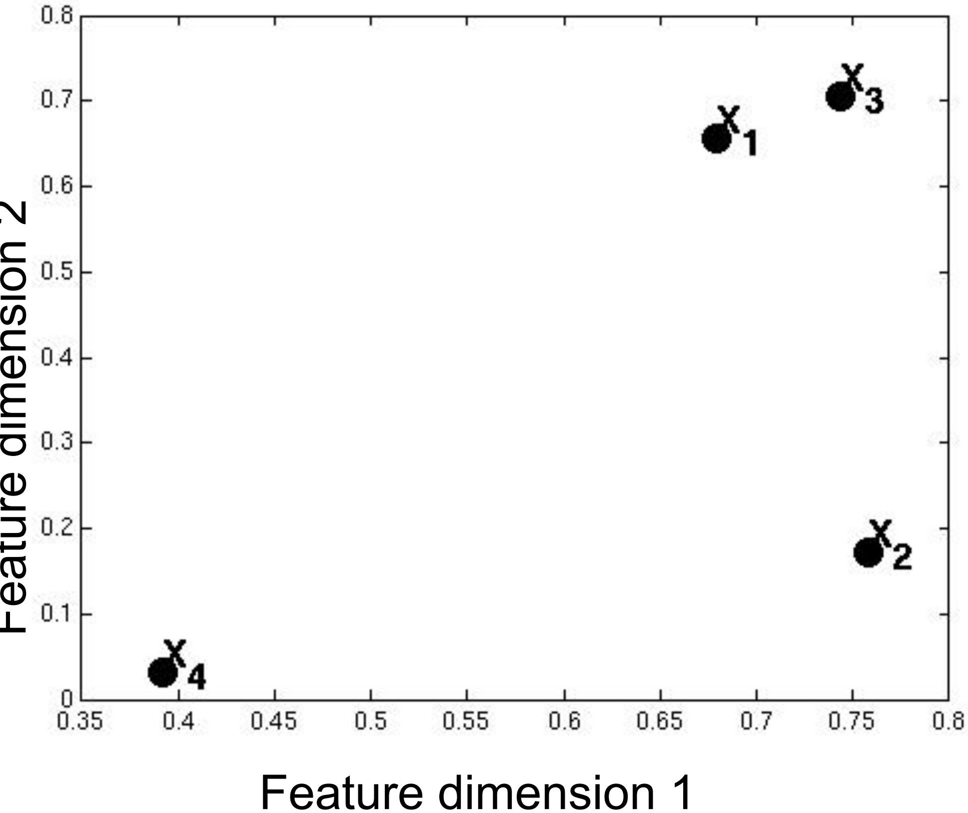
Small sigma: group only nearby points



Large sigma: group distant points

Kristen Grauman

Example: weighted graphs



- Suppose we have a 4-pixel image

  (i.e., a 2 x 2 matrix)

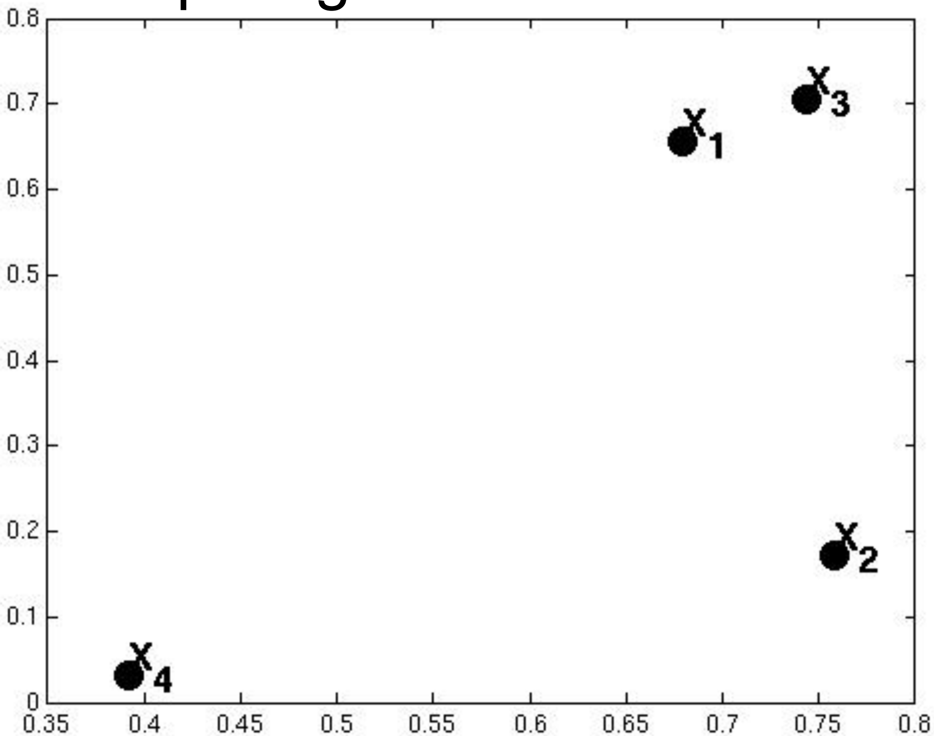
- Each pixel described by 2 features

Dimension of data points : d = 2
Number of data points : N = 4

Kristen Grauman

Example: weighted graphs

Computing the distance matrix:



D(:,1)=

(1,:)=

(0)	0.24	0.01	0.47

for i=1:N

for j=1:N

$$D(i,j) = ||x_i - x_j||^2$$

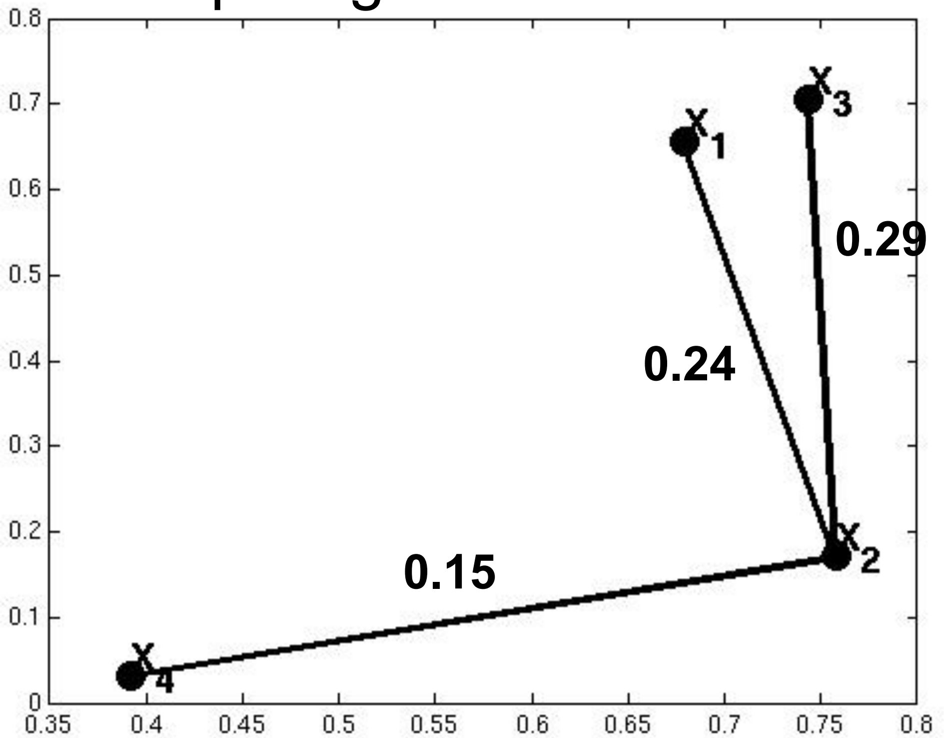
end

end

Kristen Grauman

Example: weighted graphs

Computing the distance matrix:



D(:,1)=

D(1,:)=				
	(0)	0.24	0.01	0.47

for i=1:N

for j=1:N

D(i,j) = $||x_i - x_j||^2$

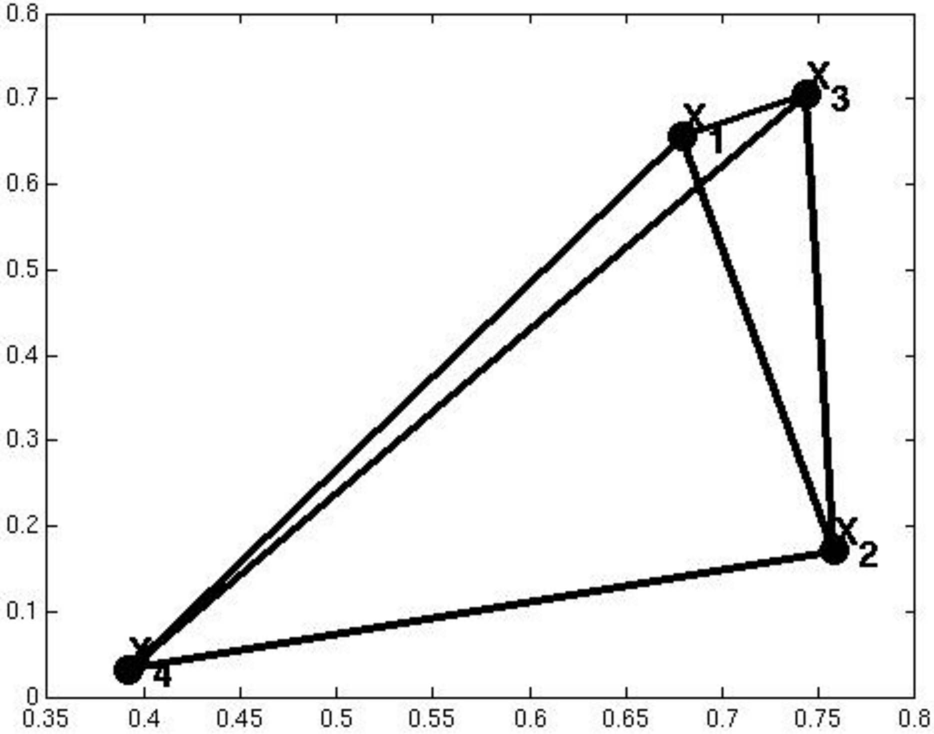
end

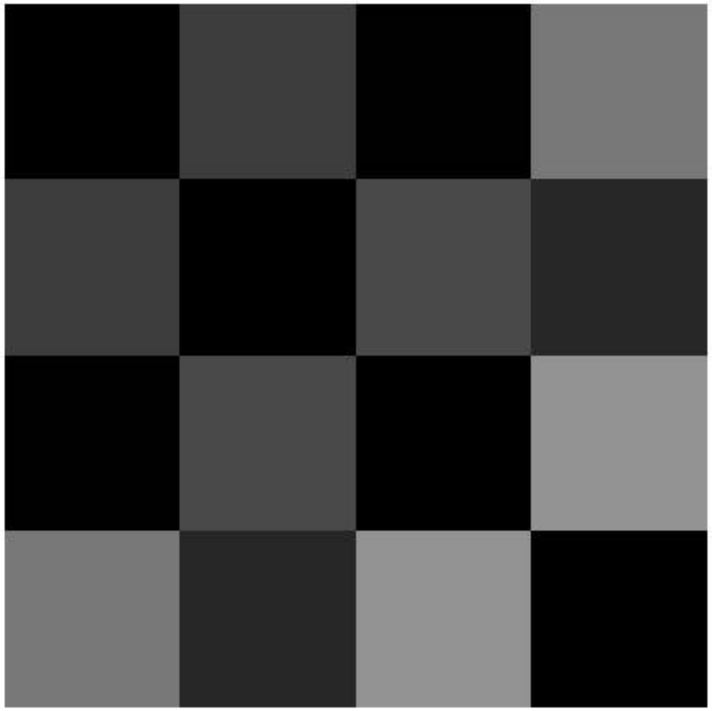
end

Kristen Grauman

Example: weighted graphs

Computing the distance matrix:





N x N matrix

for i=1:N

for j=1:N

$D(i,j) = ||x_i - x_j||^2$

end

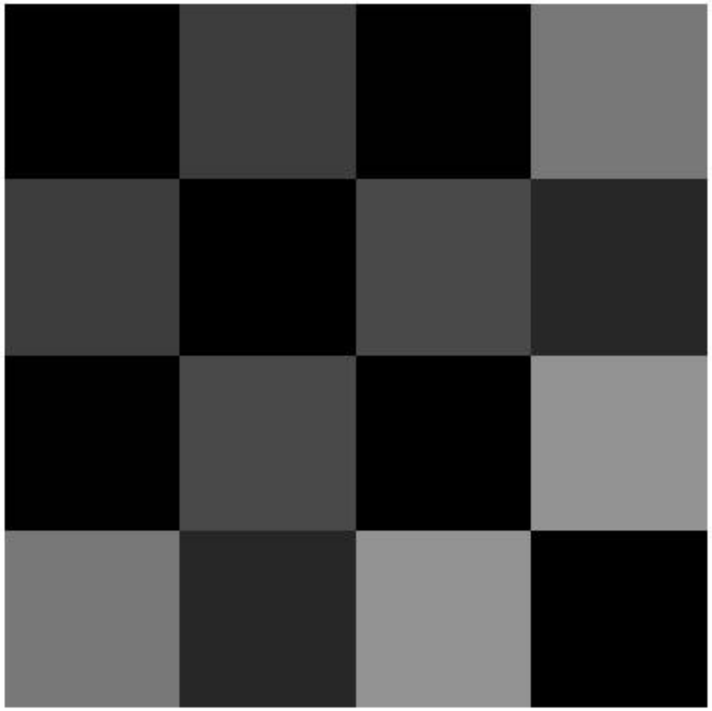
end

Kristen Grauman

Example: weighted graphs

Distances → affinities

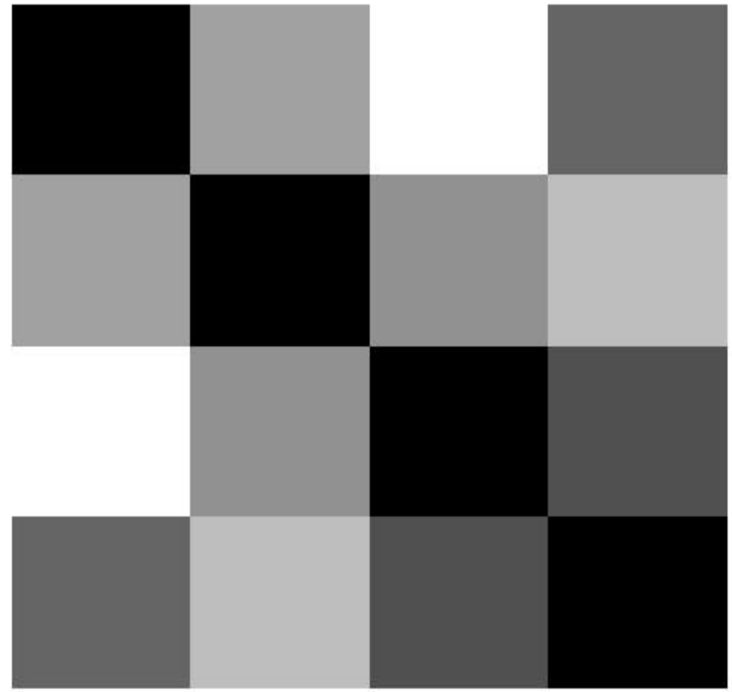
D



```
for i=1:N
  for j=1:N
    D(i,j) = ||x_i- x_j||^2
  end
end
```

A

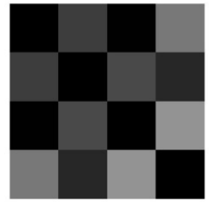
$\sigma = 0.5$



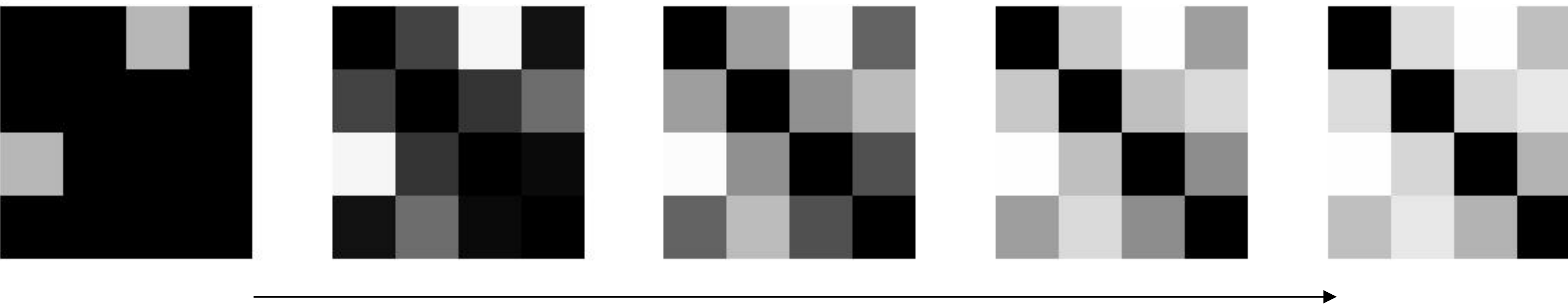
```
for i=1:N
  for j=i+1:N
    A(i,j) = exp(-1/(2*σ^2)*||x_i- x_j||^2);
    A(j,i) = A(i,j);
  end
end
```

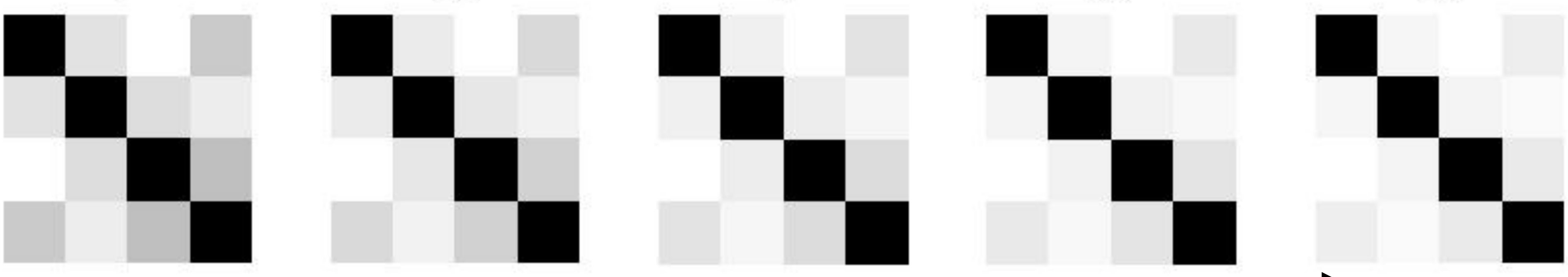

Kristen Grauman

Scale parameter σ affects affinity

Distance matrix **D=** 

Affinity matrix with increasing σ:

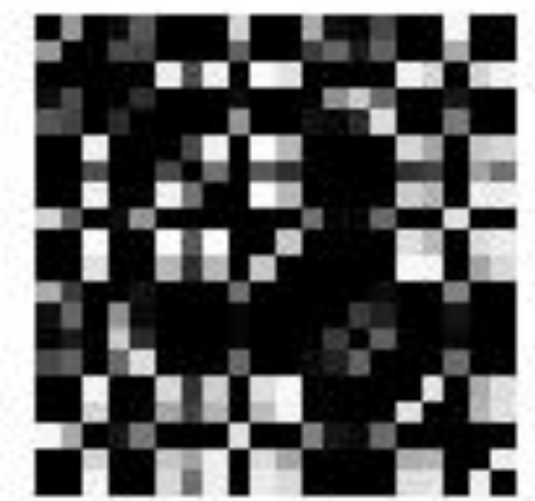


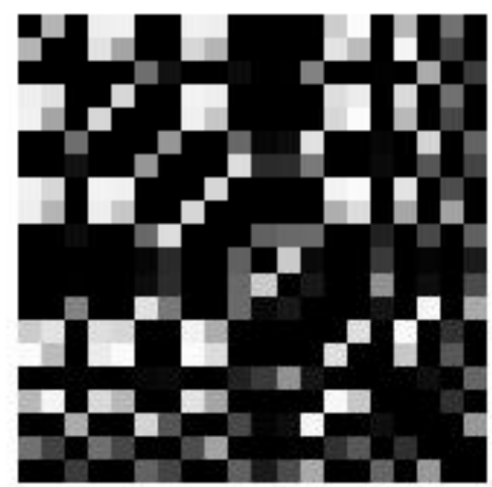


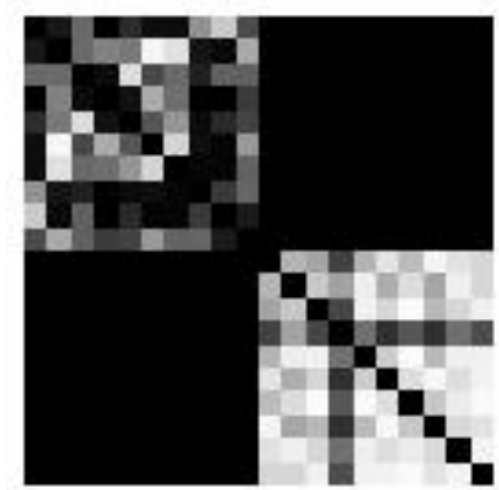
Kristen Grauman

Visualizing a shuffled affinity matrix

If we permute the order of the vertices as they are referred to in the affinity matrix, we see different patterns:



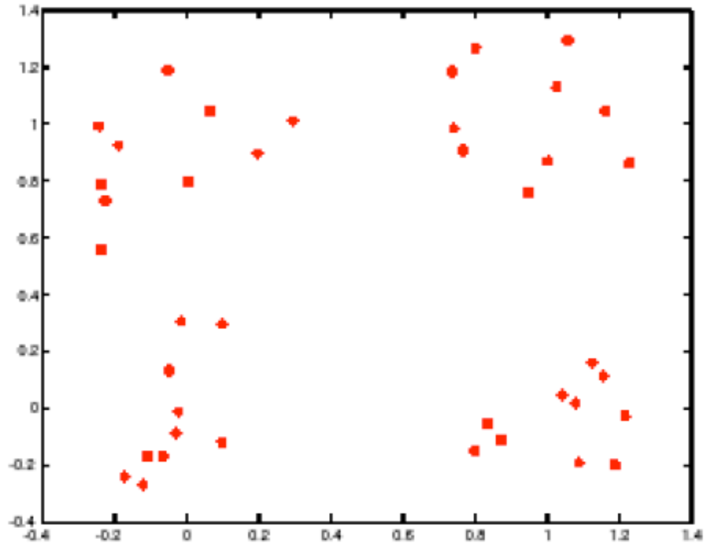




Kristen Grauman

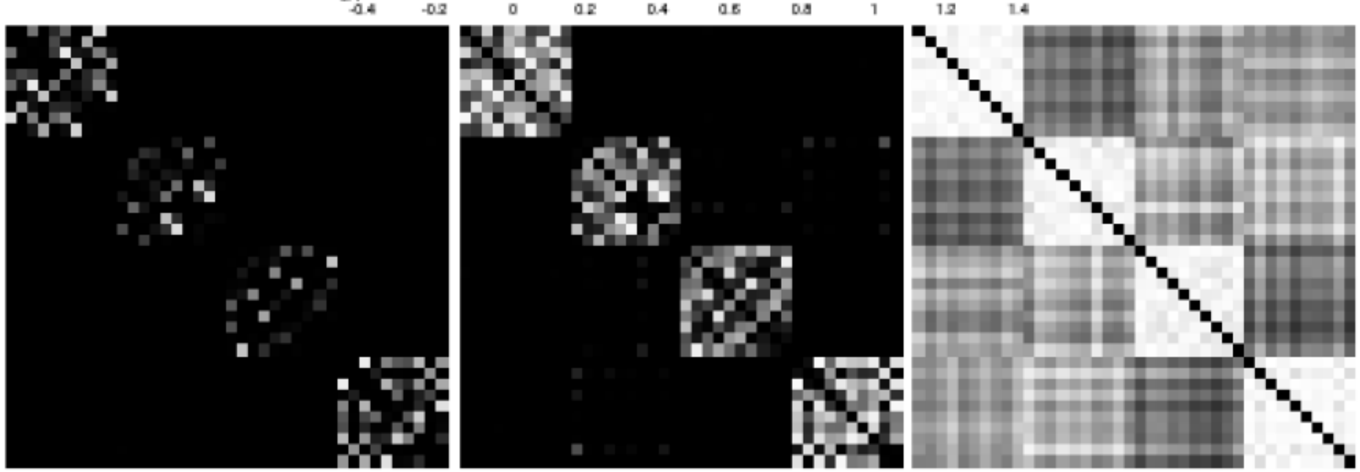
Measuring affinity

Data points



σ=.2

Affinity matrices

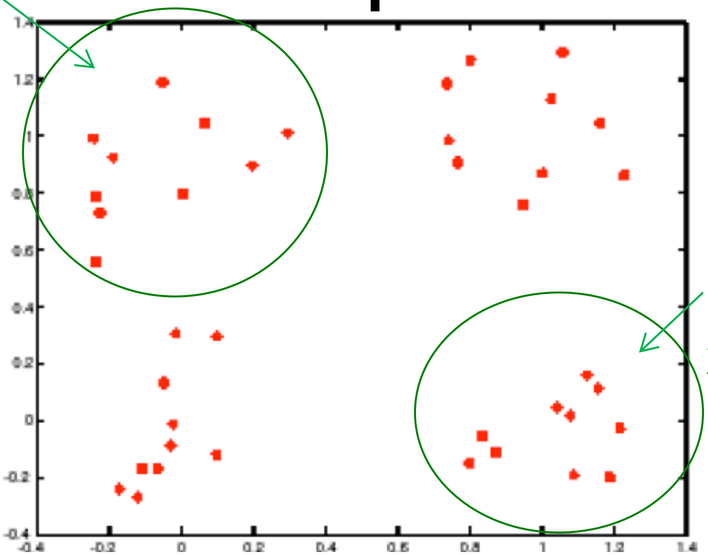


σ=.1 σ=.2 σ=1

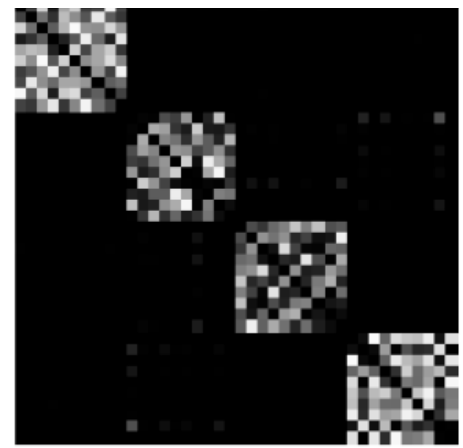
Slide credit: Kristen Grauman

Measuring affinity

40 data points



40 x 40 affinity matrix *A*

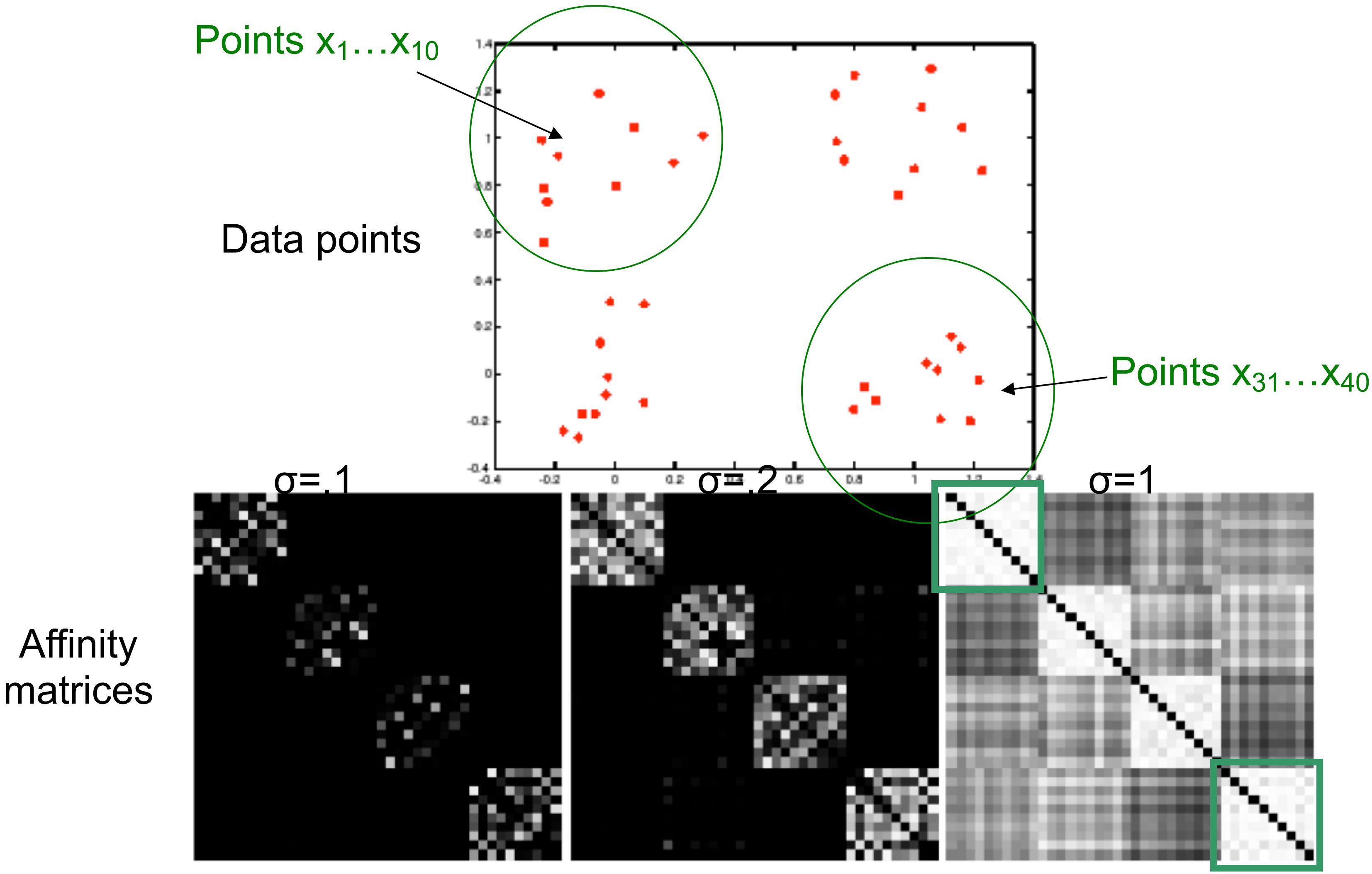


$$A(i,j) = \exp\{-(1/_{2\sigma^2})\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\}$$

1. What do the **blocks** signify?
2. What does the **symmetry** of the matrix signify?
3. How would the matrix change with **larger value of σ**?

Slide credit: Kristen Grauman

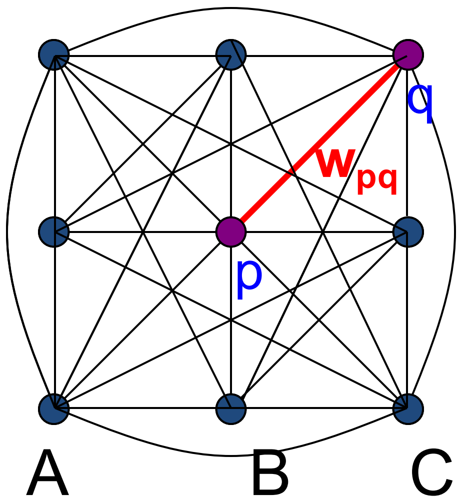
Putting it together



$$A(i,j) = \exp\{-(^1/_{2\sigma^2})\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\}$$

Kristen Grauman

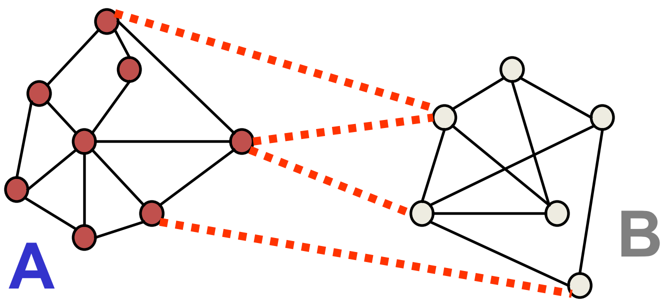
Segmentation by Graph Cuts



- Break Graph into Segments
  - Want to 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

Source: Steve Seitz

Cuts in a graph: Min cut



- Link Cut
  - set of links whose removal makes a graph disconnected
  - cost of a cut: $cut(A,B) = \sum_{p \in A, q \in B} w_{p,q}$

Find minimum cut

- gives you a segmentation
- fast algorithms exist for doing this

Source: Steve Seitz

Minimum cut

- Problem with minimum cut:

  Weight of cut proportional to number of edges in the cut; tends to produce small, isolated components.

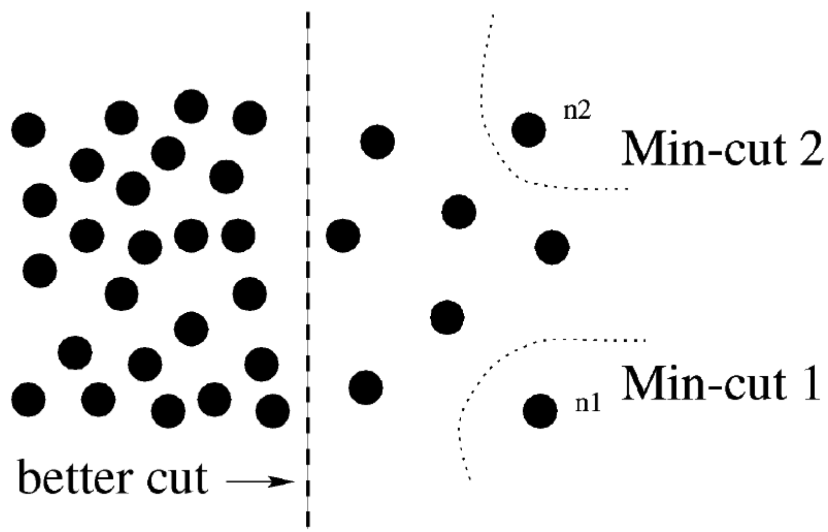
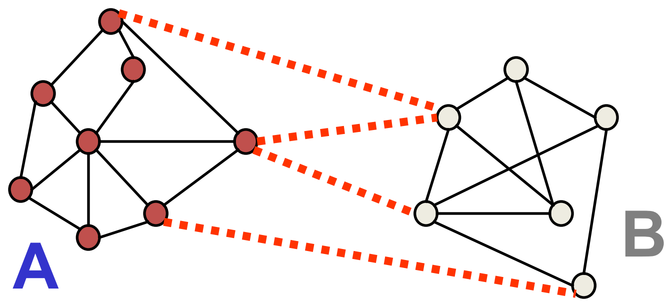


Fig. 1. A case where minimum cut gives a bad partition.

[Shi & Malik, 2000 PAMI]

Slide credit: Kristen Grauman

Cuts in a graph: Normalized cut



Normalized Cut

- fix bias of Min Cut by **normalizing** for size of segments:

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

assoc(A,V) = sum of weights of all edges that touch A

- Ncut value small when we get two clusters with many edges with high weights, and few edges of low weight between them
- Approximate solution for minimizing the Ncut value : generalized eigenvalue problem.

Source: Steve Seitz

J. Shi and J. Malik, Normalized Cuts and Image Segmentation, CVPR, 1997

Example results

Slide credit: Kristen Grauman

Results: Berkeley Segmentation Engine

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

Slide credit: Kristen Grauman

Normalized cuts: pros and cons

Pros:

- Generic framework, flexible to choice of function that computes weights ("affinities") between nodes
- Does not require model of the data distribution

Cons:

- Time complexity can be high
  - Dense, highly connected graphs → many affinity computations
  - Solving eigenvalue problem
- Preference for balanced partitions

Kristen Grauman

Grouping in Vision

Segmentation as Clustering

Mode finding & Mean-Shift

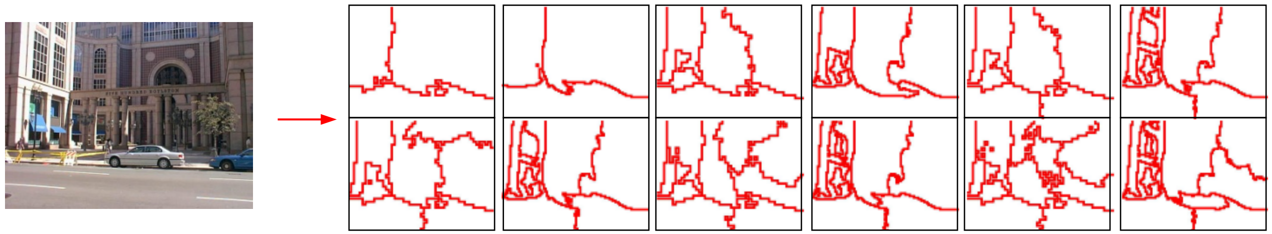
Graph-Based Algorithms

$\boxed{\text{Segments as Primitives}}$

CNN-Based Approaches

Segments as primitives for recognition

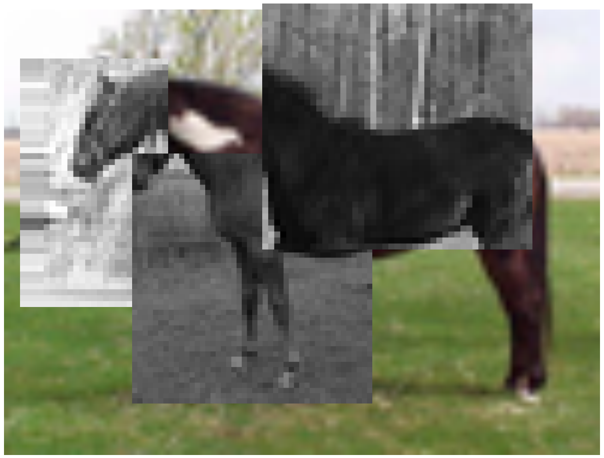
Multiple segmentations

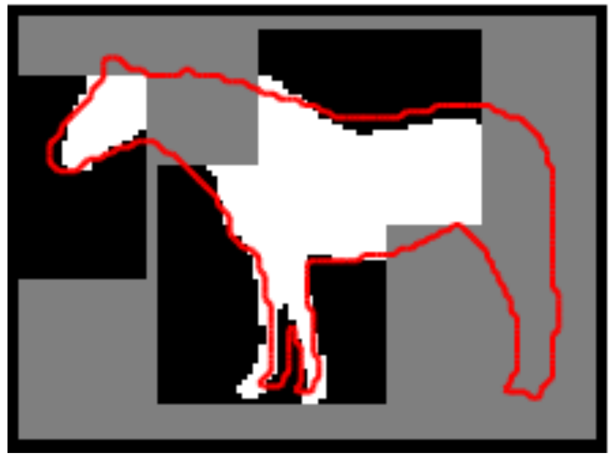


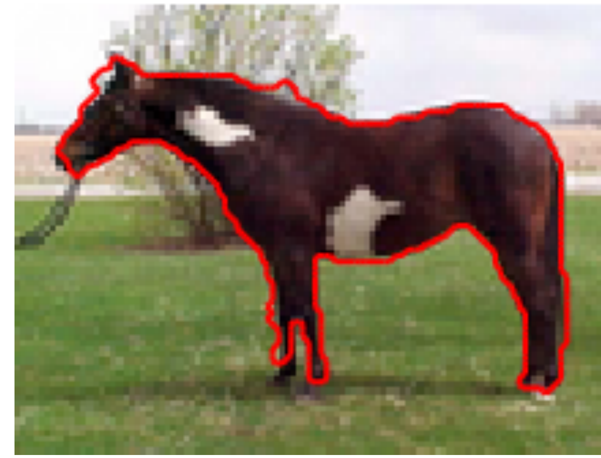
- B. Russell et al., [“Using Multiple Segmentations to Discover Objects and their Extent in Image Collections,”]() CVPR 2006

Slide credit: Lana Lazebnik

Top-down segmentation







E. Borenstein and S. Ullman, "Class-specific, top-down segmentation," ECCV 2002

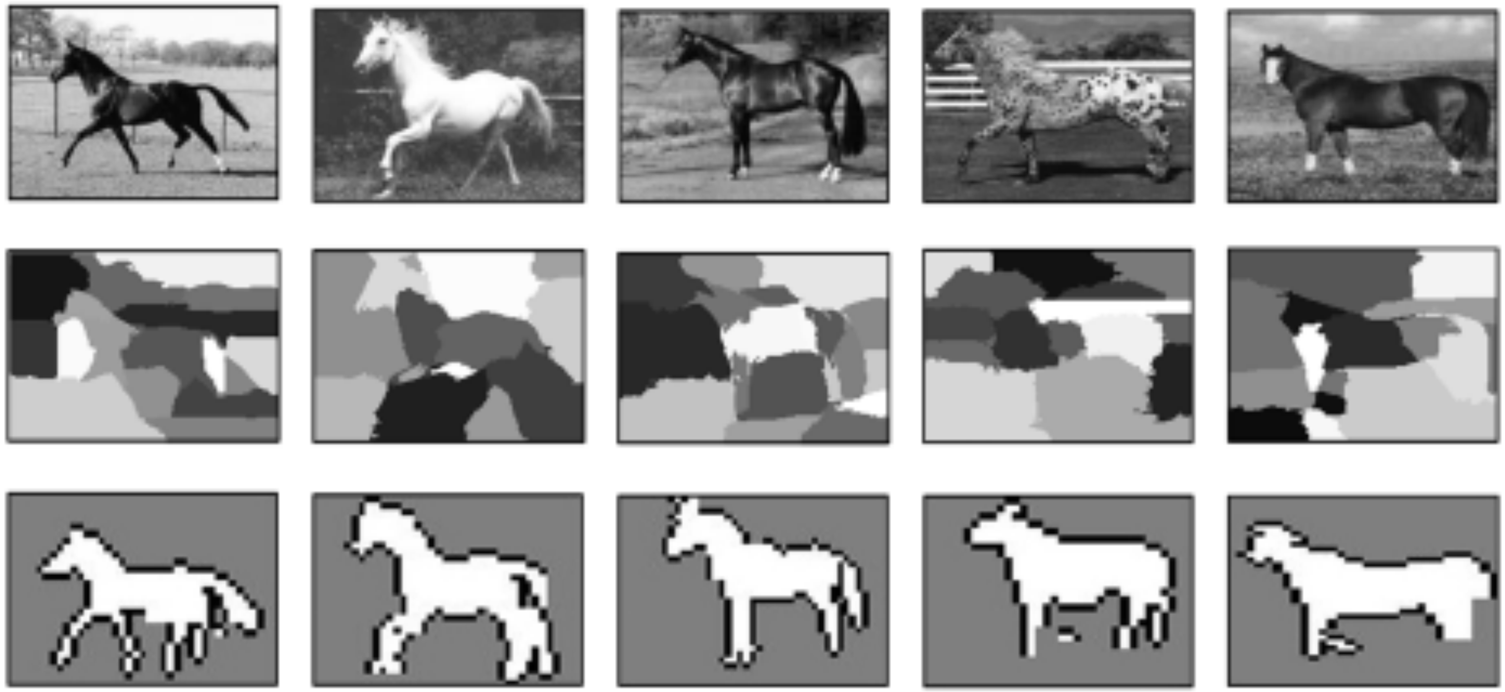
A. Levin and Y. Weiss, "Learning to Combine Bottom-Up and Top-Down Segmentation," ECCV 2006.

Slide credit: Lana Lazebnik

Top-down segmentation

Normalized cuts

Top-down segmentation



- E. Borenstein and S. Ullman, [“Class-specific, top-down segmentation,”]() ECCV 2002
- A. Levin and Y. Weiss, [“Learning to Combine Bottom-Up and Top-Down Segmentation,”]() ECCV 2006.

Slide credit: Lana Lazebnik

Summary

- Segmentation to find object boundaries or mid-level regions, tokens.
- Bottom-up segmentation via clustering
  - General choices -- features, affinity functions, and clustering algorithms
- Grouping also useful for quantization, can create new feature summaries
  - Texton histograms for texture within local region
- Example clustering methods
  - K-means
  - Mean shift
  - Graph cut, normalized cuts

Slide credit: Kristen Grauman

Grouping in Vision

Segmentation as Clustering

Mode finding & Mean-Shift

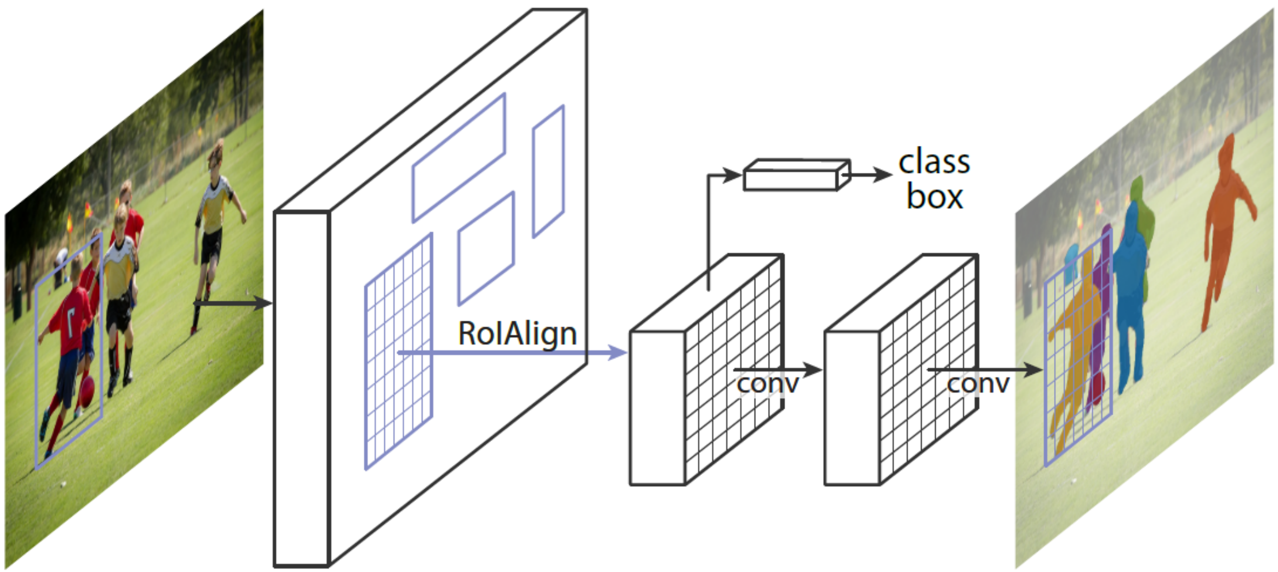
Graph-Based Algorithms

Segments as Primitives

CNN-Based Approaches

More recently…

- Neural networks to learn both local feature affinities and top-down context



- He et al., [“Mask R-CNN,”]() ICCV 2017 (Best paper)

More recently…

- Segmenting both classes and instances

- He et al., “Mask R-CNN,” ICCV 2017 (Best paper)