CENG 483 - Intro. to Computer Vision

Clustering, segmentation & k-means clustering

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

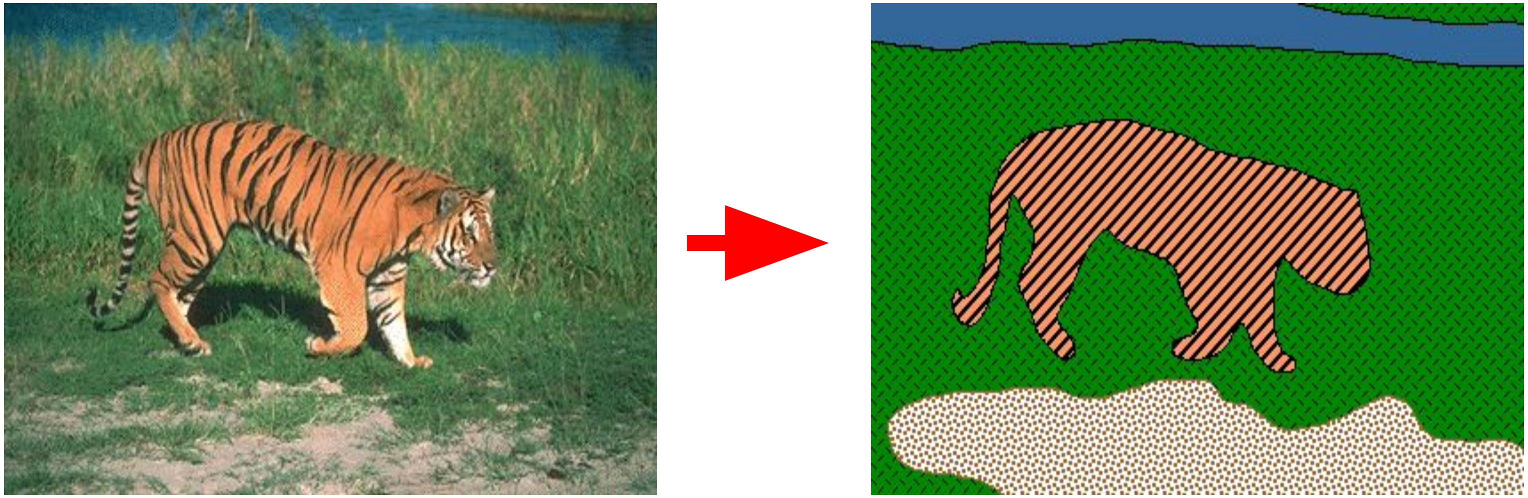
What we will learn today

- **Introduction to segmentation and clustering**
- Gestalt theory for perceptual grouping
- K-means clustering

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Image Segmentation (unsupervised)

- A very commonly studied **grouping problem**
- Goal: identify groups of pixels that go together



Slide credit: Steve Seitz, Kristen Grauman

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Segmentation vs Semantic Segmentation

Typically:

- **Image (unsupervised) segmentation**: unsupervised, grouping based
  - This is what we mean throughout this lecture
- **Semantic segmentation**: supervised, category-level

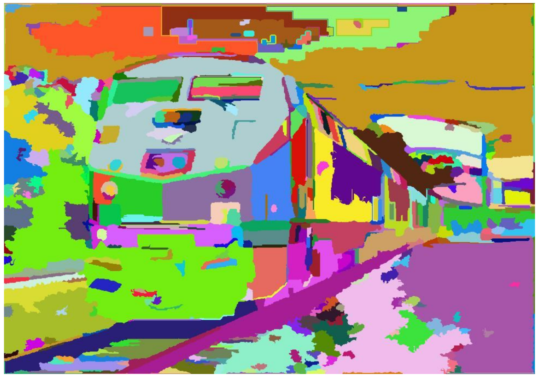
The Goals of Segmentation

- **Example application #1** Separate image into coherent "objects"

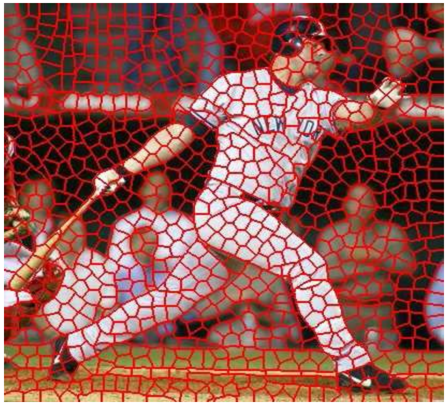
https://arxiv.org/pdf/2102.06191.pdf

The Goals of Segmentation

- **Example application #2** Group similar pixels for efficiency of further processing (superpixels, etc)



[Felzenszwalb and Huttenlocher 2004]



[Hoiem et al. 2005, Mori 2005]

[Shi and Malik 2001]

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

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Slide: Derek Hoiem

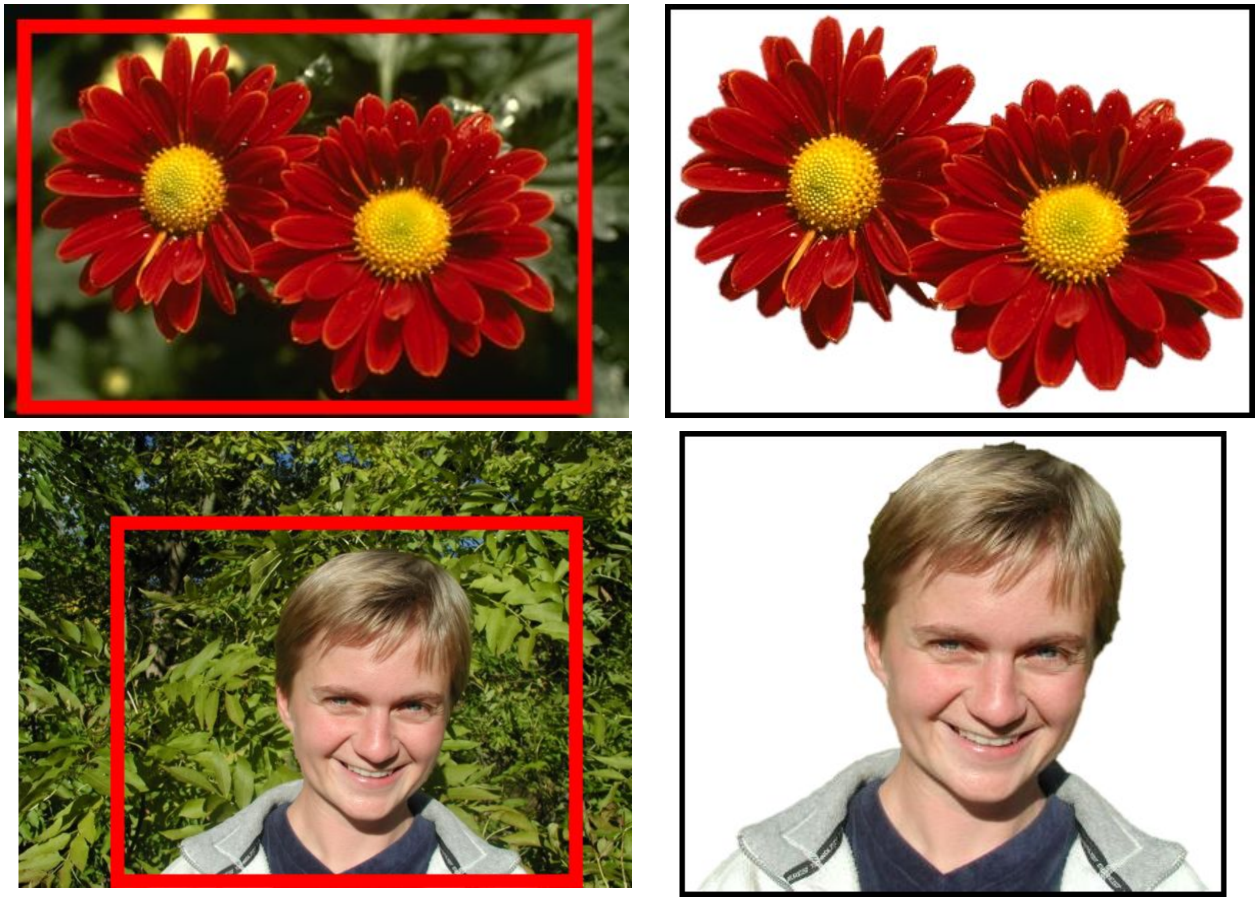
The Goals of Segmentation

- **Example application #3** Better feature support

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

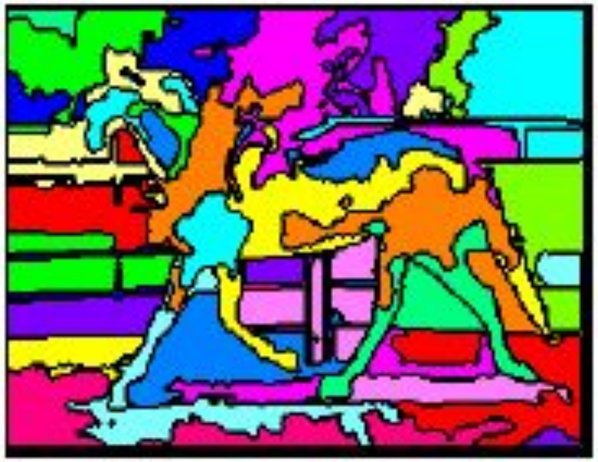
The Goals of Segmentation

- **Example application #4** Image editing

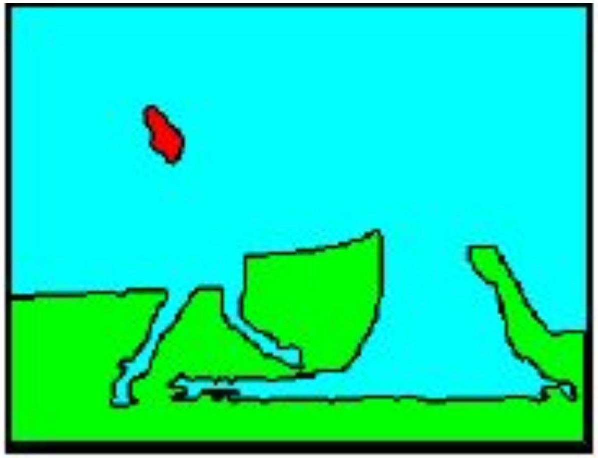


Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

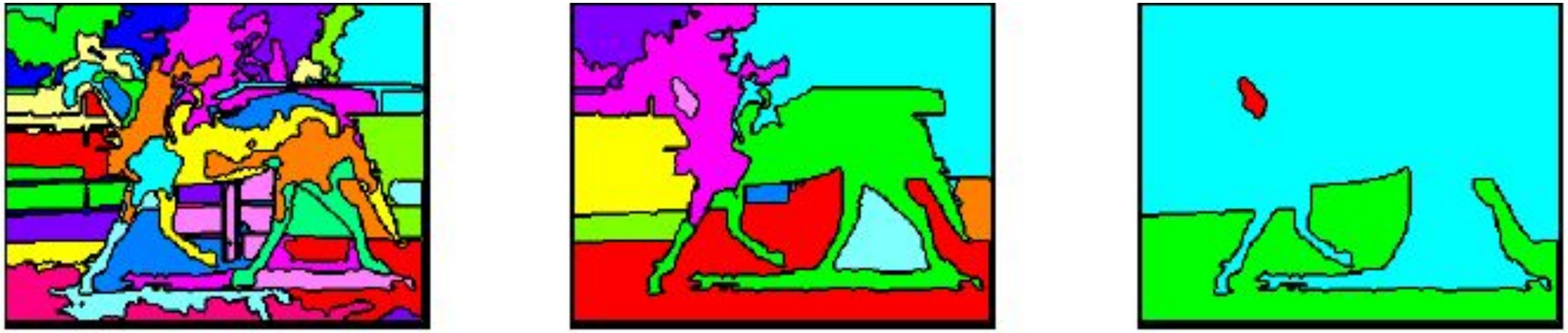
Types of segmentations



Oversegmentation



Undersegmentation



Multiple Segmentations

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

One way to think about "segmentation" is Clustering

Clustering: group together similar data points and represent them with a single token

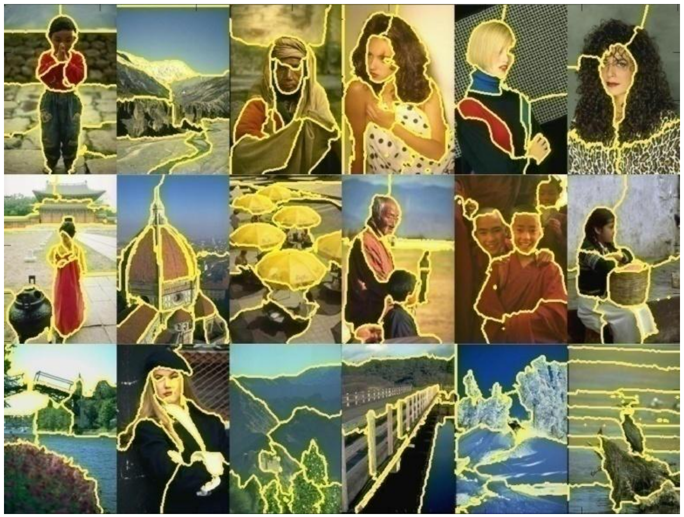
Key Challenges:

1) What makes two points/images/patches similar?

2) How do we compute an overall grouping from pairwise similarities?

Slide: Derek Hoiem

There are many other grouping tasks in Vision



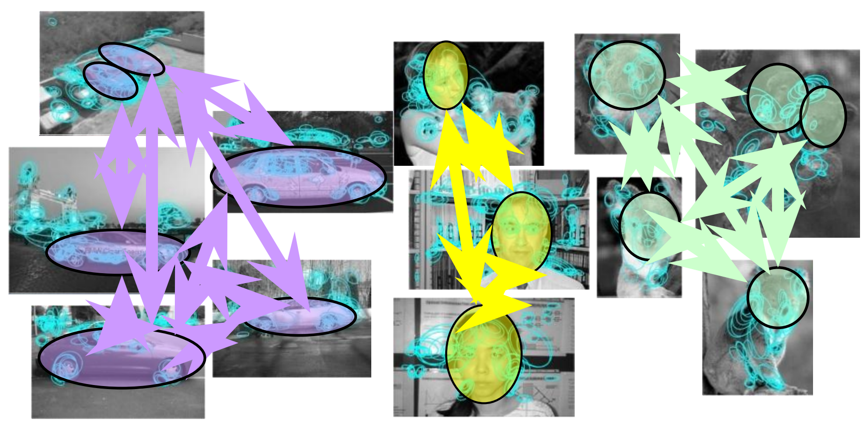
Determining image regions

Grouping video frames into shots

Figure-ground

What things should be grouped?

What cues indicate groups?



Object-level grouping

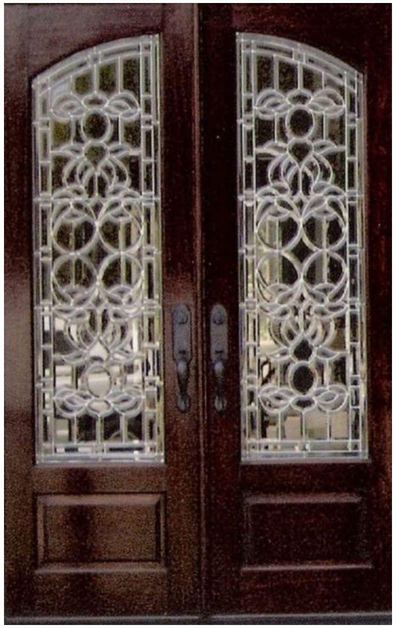
Original version by: Kristen Grauman

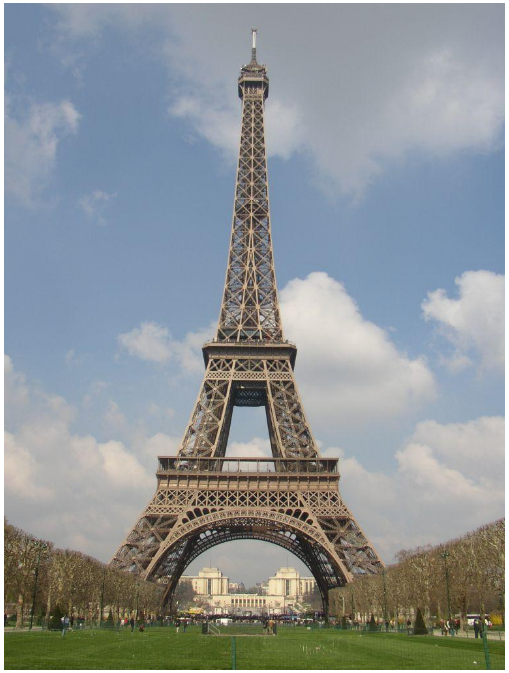
Similarity

Slide credit: Kristen Grauman

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Symmetry





Slide credit: Kristen Grauman

Common Fate

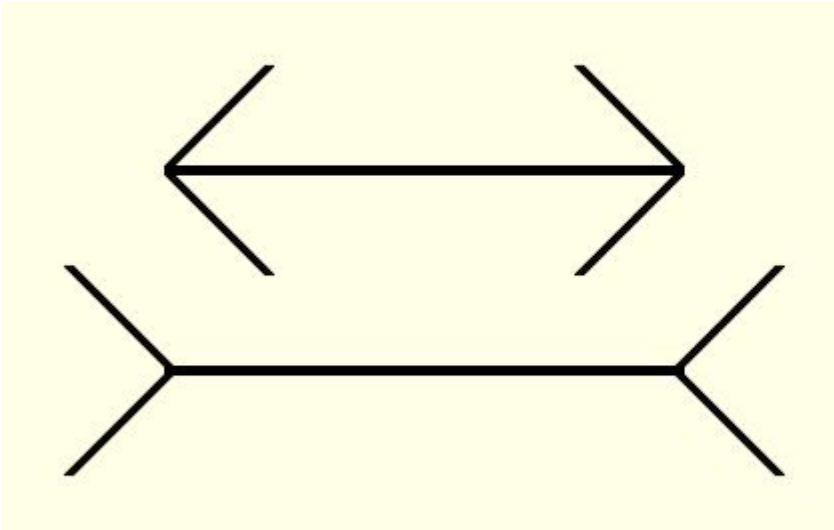
Image credit: Arthus-Bertrand (via F. Durand)

Slide credit: Kristen Grauman

Proximity

Slide credit: Kristen Grauman

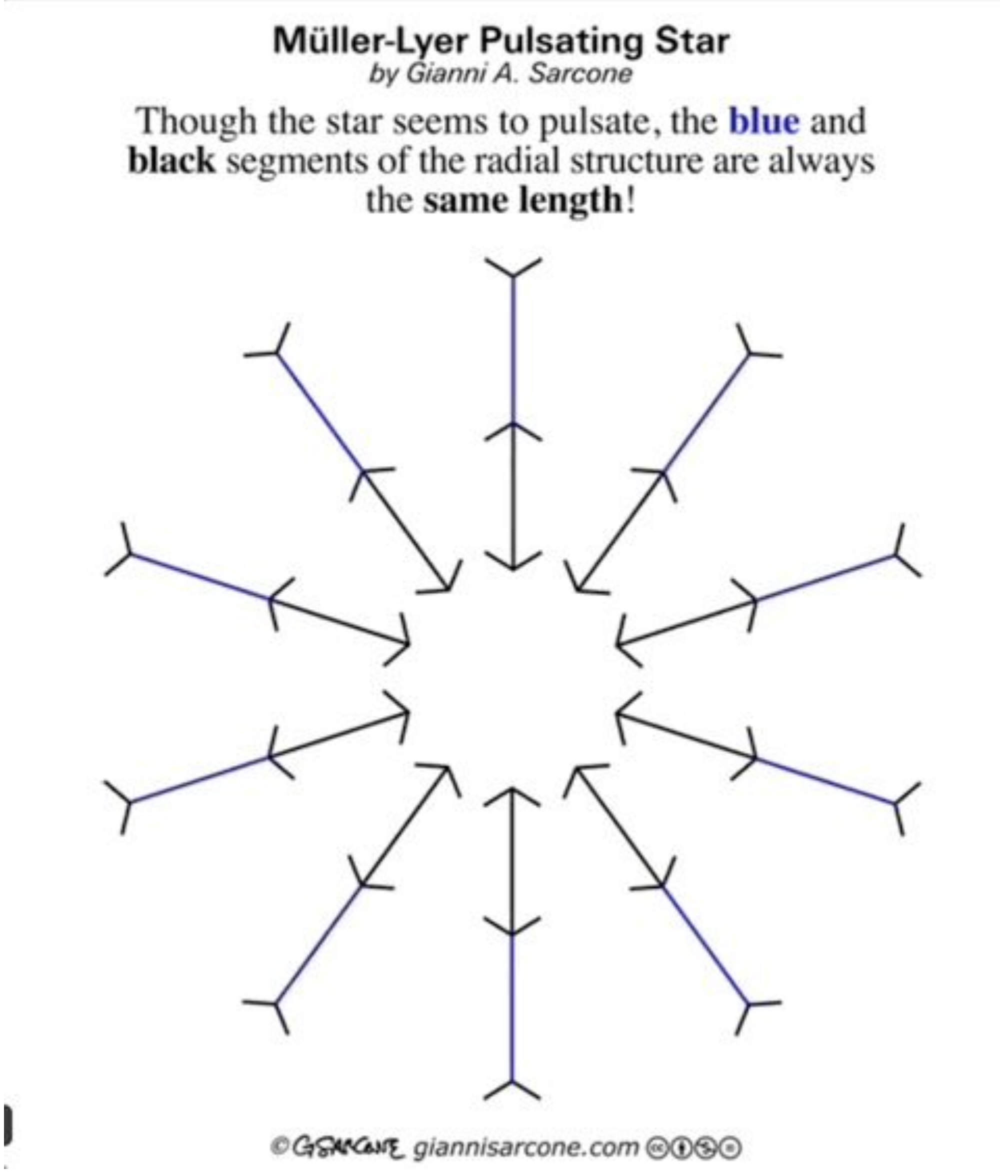
Muller-Lyer Illusion



- What makes the bottom line look longer than the top line? (Answer is not fully clear)

See for potential explanations, e.g. misleading 3D cues:
https://en.wikipedia.org/wiki/M%C3%BCller-Lyer_illusion

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna



https://twitter.com/social_brains/status/972325994391486464

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

What we will learn today

- Introduction to segmentation and clustering
- **Gestalt theory for perceptual grouping**
- K-means clustering

The Gestalt School

- *A school of psychology that emerged in the early twentieth century in Austria and Germany as a theory of perception.*

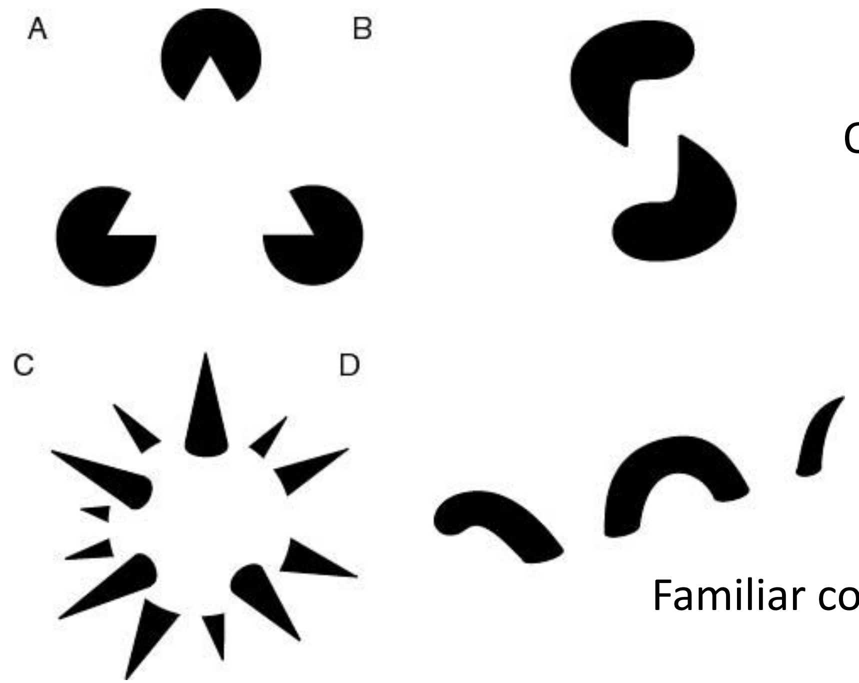
- gestaltism = configurationism

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

The Gestalt School

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

Illusory/subjective contours



Occlusion

Familiar configuration

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

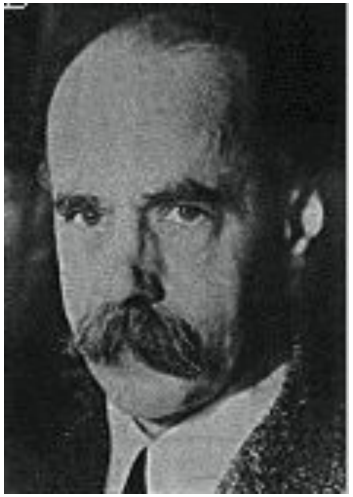
Slide credit: Svetlana Lazebnik

Gestalt Theory

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

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

Max Wertheimer
(1880-1943)



Untersuchungen zur Lehre von der Gestalt,
***Psychologische Forschung*, Vol. 4, pp. 301-350, 1923**
http://psy.ed.asu.edu/~classics/Wertheimer/Forms/forms.htm

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Gestalt Factors

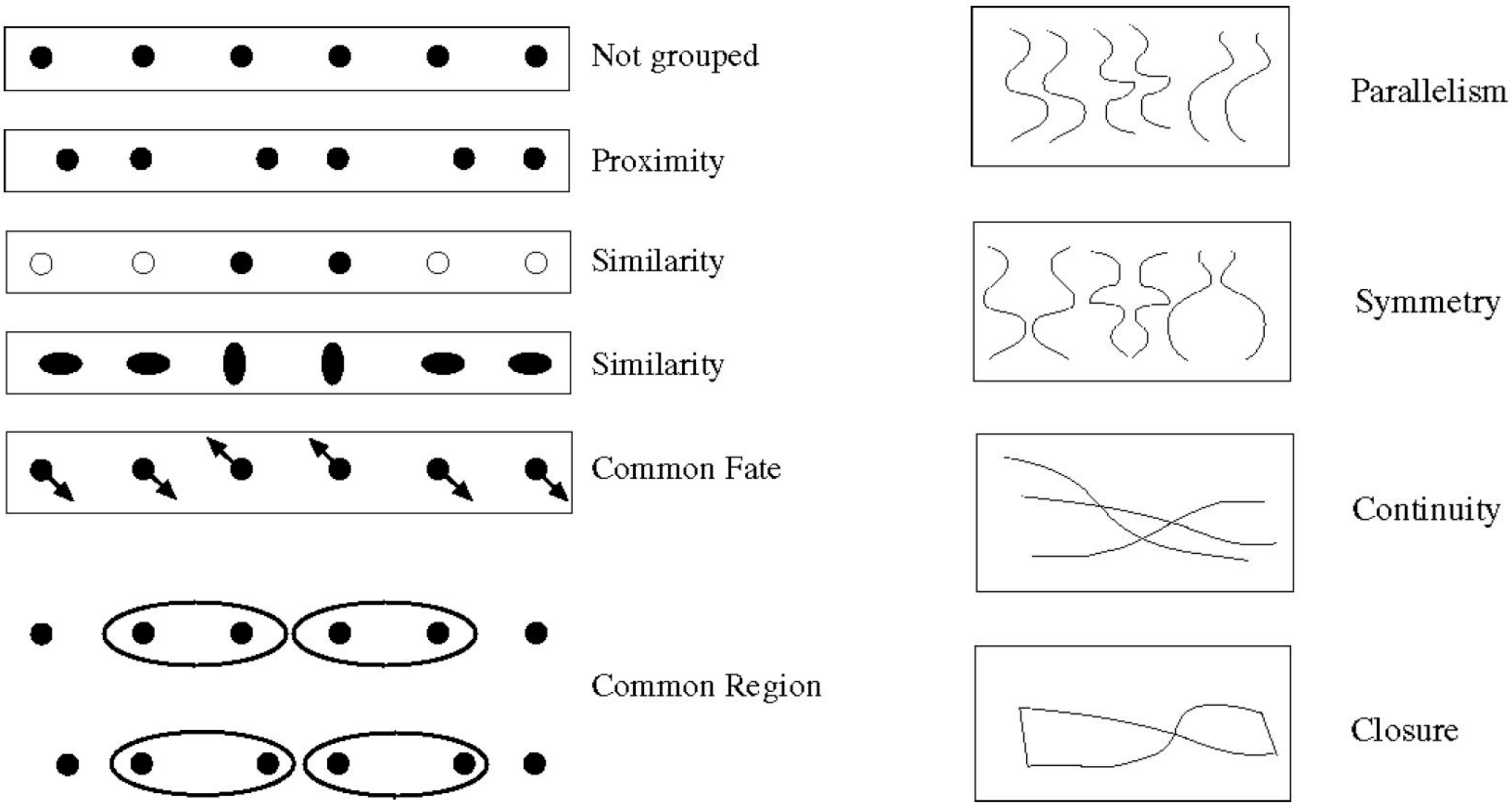
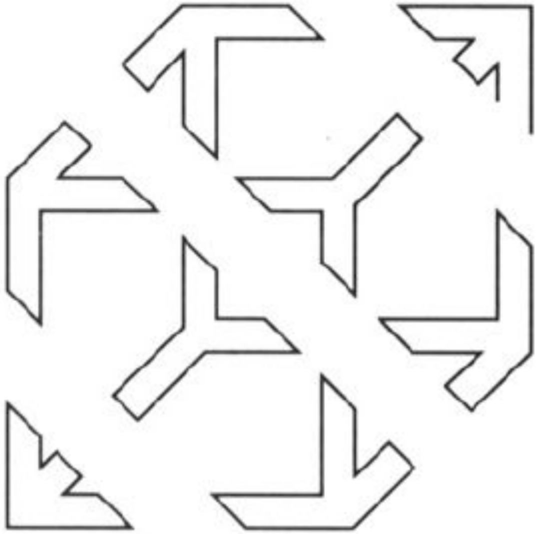


Image source: Forsyth & Ponce

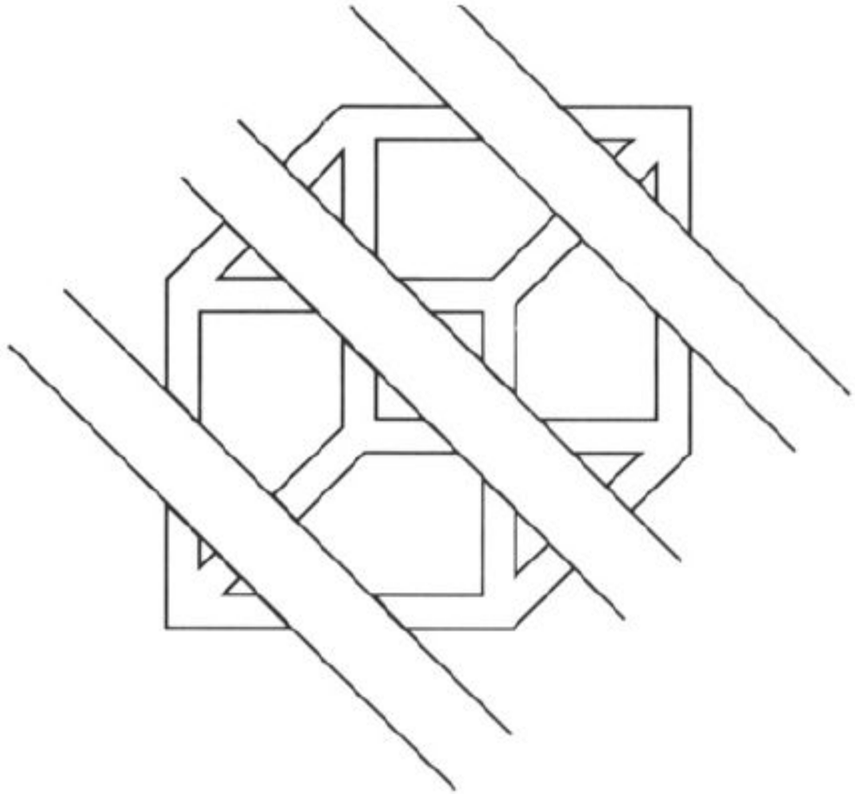
- These factors make intuitive sense, but are very difficult to translate into algorithms.

Continuity through Occlusion Cues



Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Continuity through Occlusion Cues



Continuity, explanation by occlusion

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Continuity through Occlusion Cues

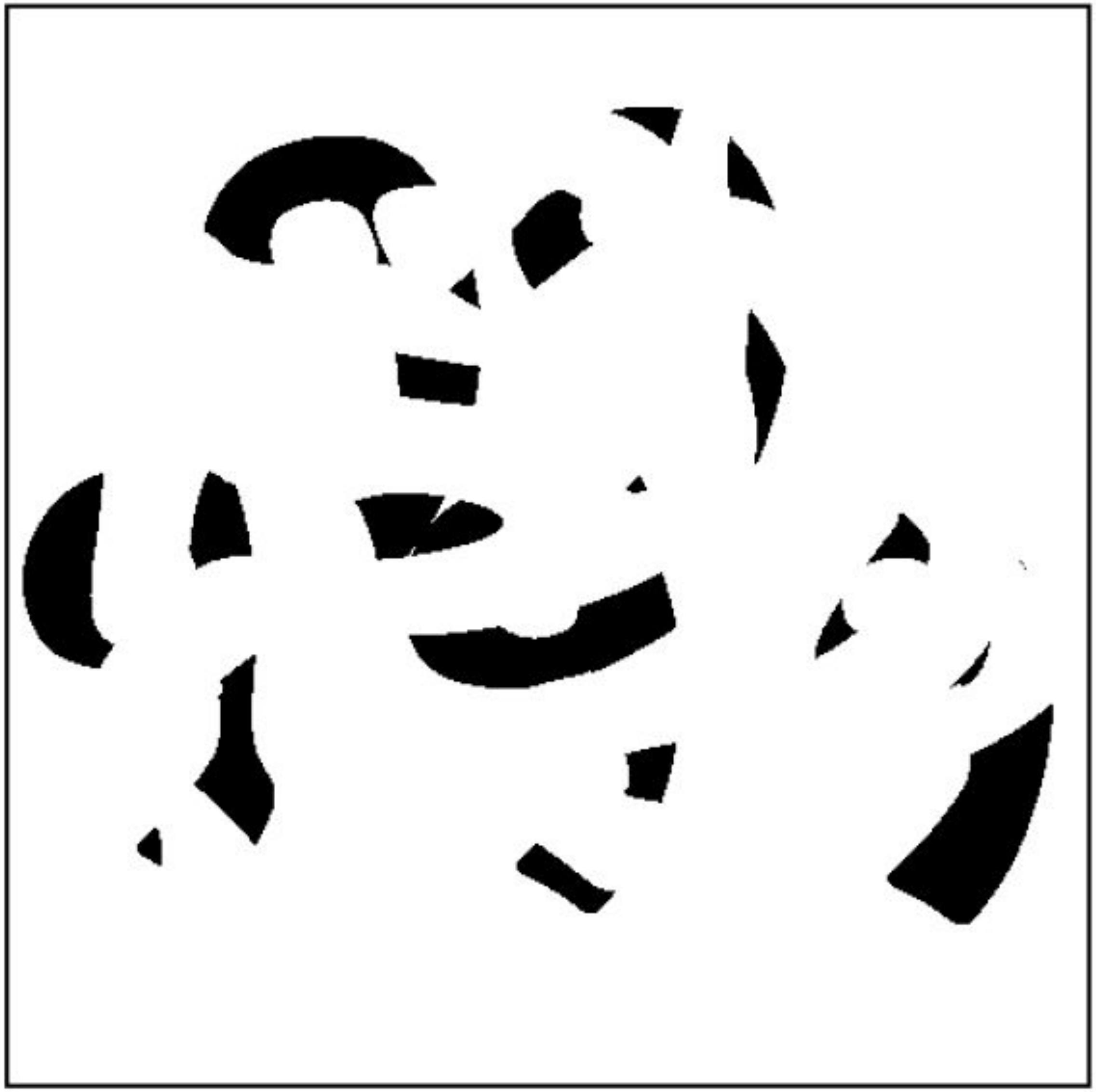


Image source: Forsyth & Ponce

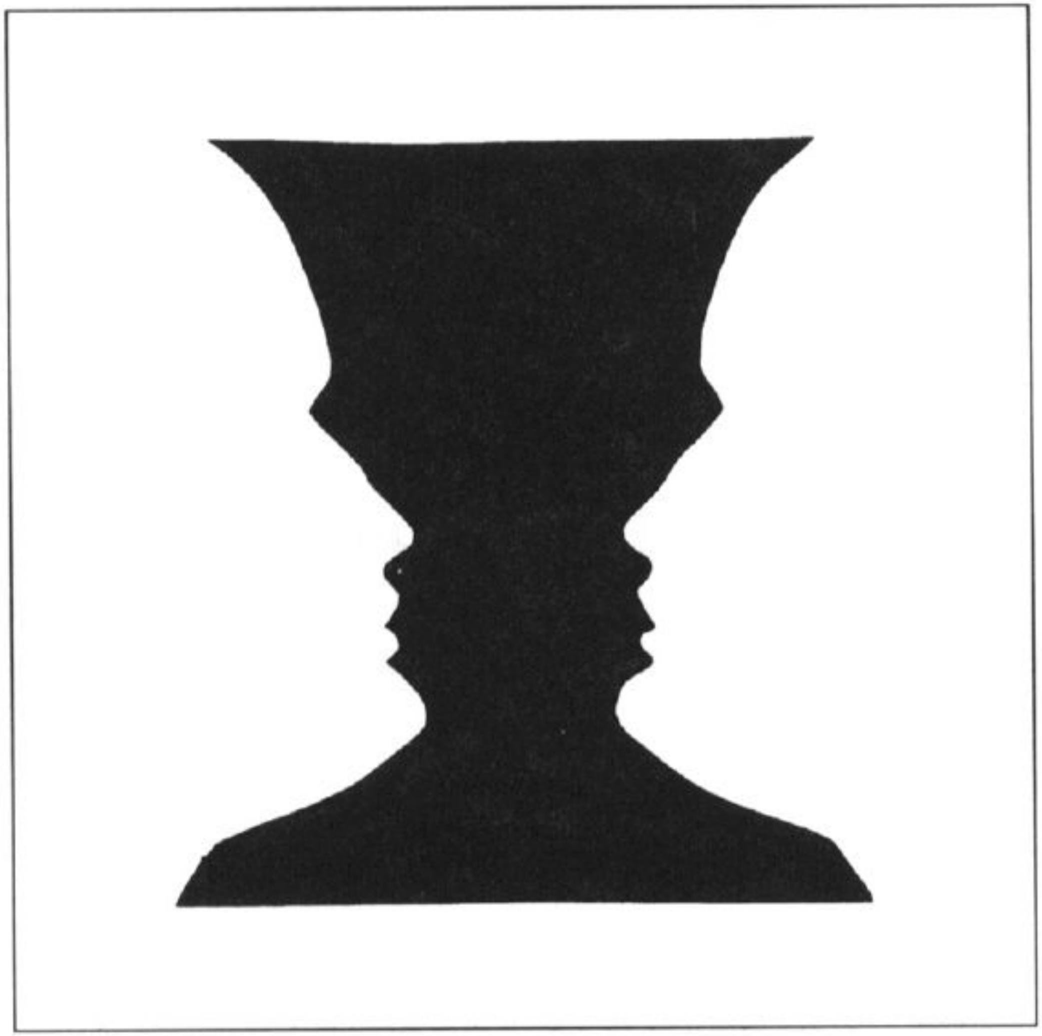
Adapted from slides by Ju

Continuity through Occlusion Cues

Image source: Forsyth & Ponce

Adapted from slides by

Figure-Ground Discrimination



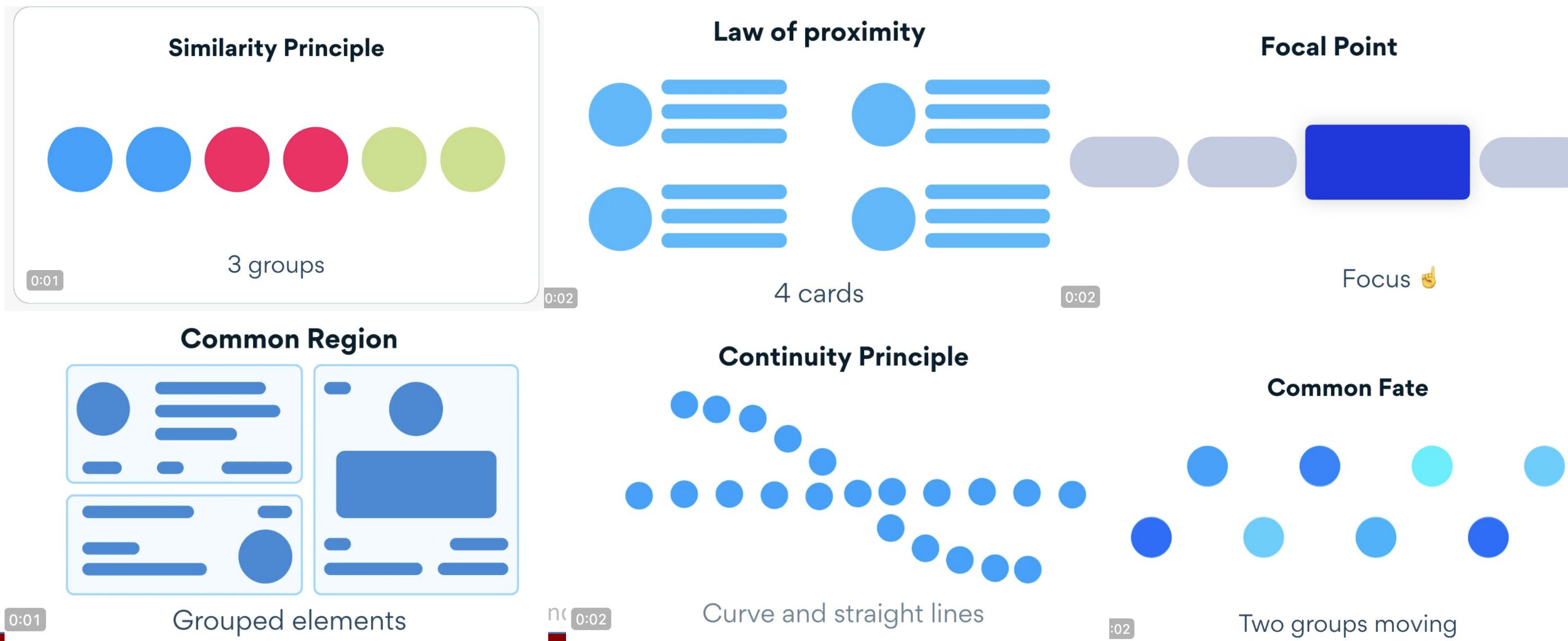
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

The Ultimate Gestalt?

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Side-topic: Gestalt Principles in Presentations

- https://twitter.com/pablostanley/status/974303621092225024



What we will learn today

- Introduction to segmentation and clustering
- Gestalt theory for perceptual grouping
- **K-means clustering**

Why do we cluster?

- **Summarizing data**
  - Look at large amounts of data
  - Patch-based compression or denoising
  - Represent a large continuous vector with the cluster number

- **Counting**
  - Histograms of texture, color, SIFT vectors

- **Segmentation**
  - Separate the image into different regions

- **Prediction**
  - Images in the same cluster may have the same labels

Slide: Derek Hoiem

General ideas

- Tokens
  - whatever we need to group (pixels, points, surface elements, etc.)
- Bottom up clustering
  - tokens belong together because they are locally coherent
- Top down clustering
  - tokens belong together because they lie on the same visual entity (object, scene…)
- Model based clustering
  - Based on fitting a model to the data set

How do we cluster?

Examples

- Agglomerative clustering (we'll skip)
  - Start with each point as its own cluster and iteratively merge the closest clusters
- K-means (we'll cover)
  - Iteratively re-assign points to the nearest cluster center
- Mean-shift clustering (we'll skip)
  - Estimate modes of pdf

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Clustering's technical elements

Clustering is an unsupervised learning method. Given items $x_1, \ldots, x_n \in \mathbb{R}^D$, the goal is to group them into clusters.

Dependending on the algorithm we need to make assumptions / take inputs, such as:
- a pairwise distance/similarity function
- the desired number of clusters
- thresholds on intra/inter cluster distance limits
- a data distribution model (eg. mixture of gaussian)

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Defining Similarity / Distance Measures

Some algorithms, such as agglomerative clustering, explicitly use a similarity or distance function.

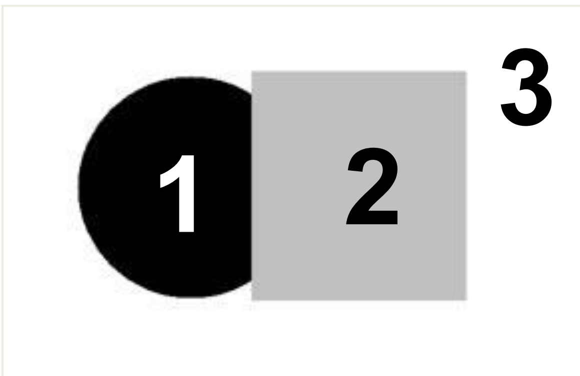
Some others, eg. k-means, implicitly assume a distance metric through the distribution assumption.

Example similarity & distance functions:

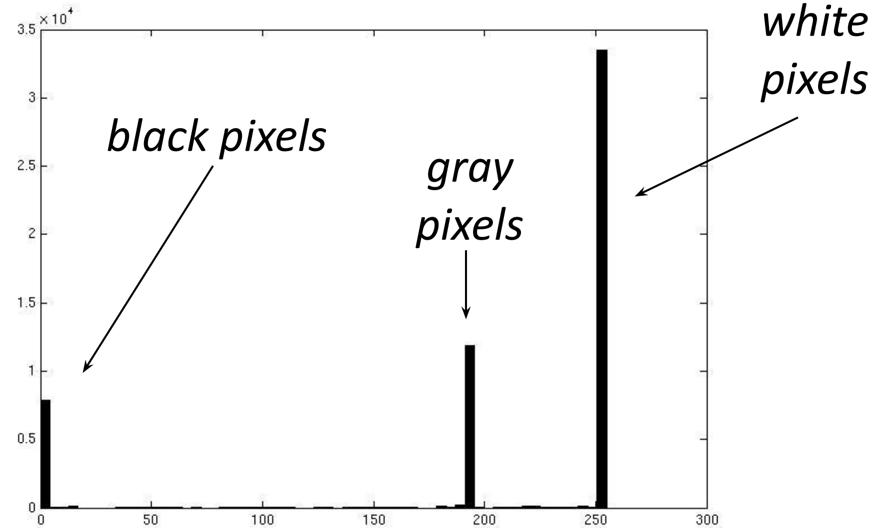
$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

$$\text{cosine distance} = D_C(A, B) := 1 - S_C(A, B)$$

Image Segmentation: Toy Example



input image

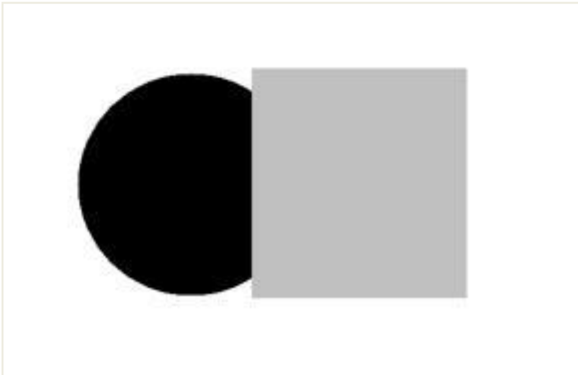


intensity

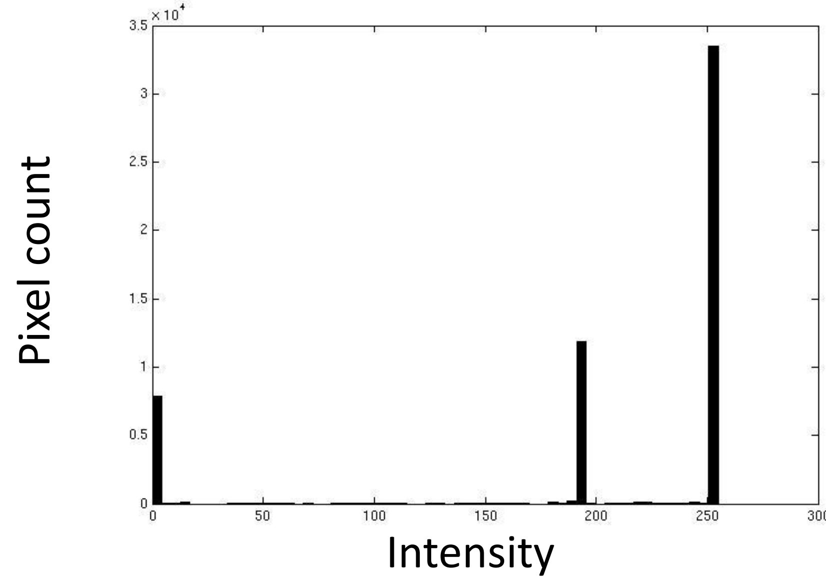
- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., segment the image based on the intensity feature.
- What if the image isn't quite so simple?

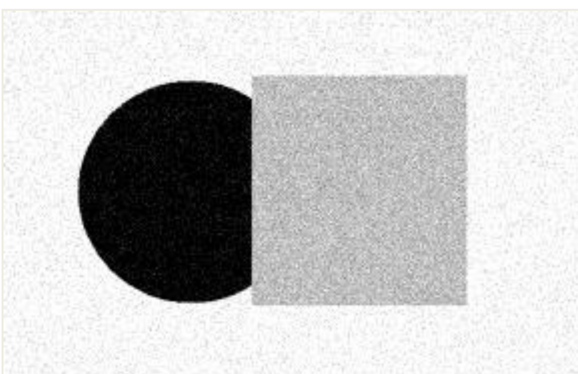
Slide credit: Kristen Grauman

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

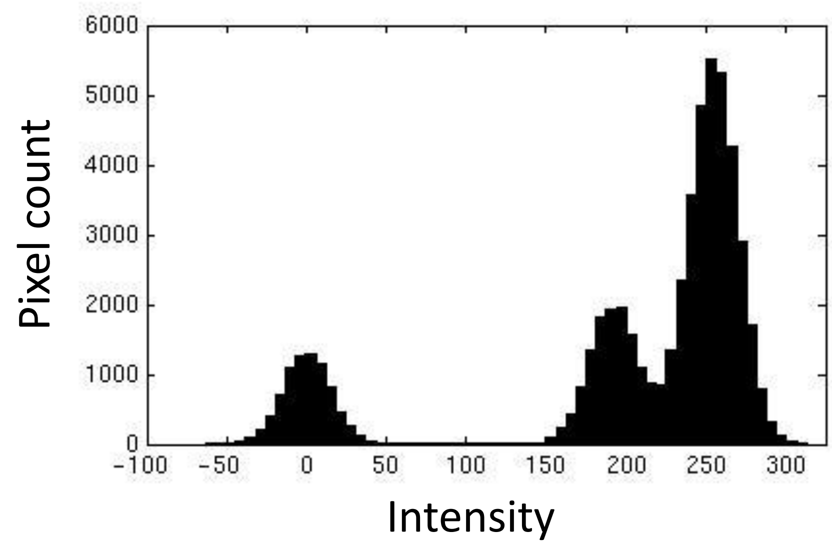


Input image



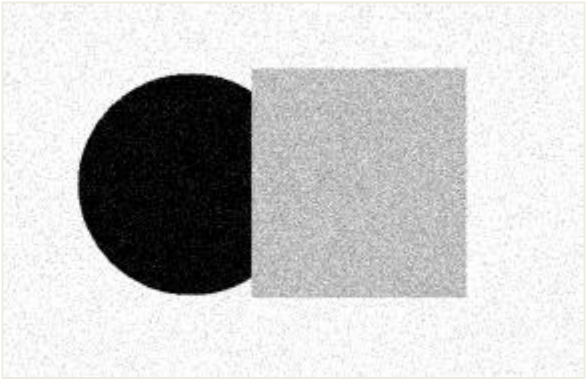


Input image

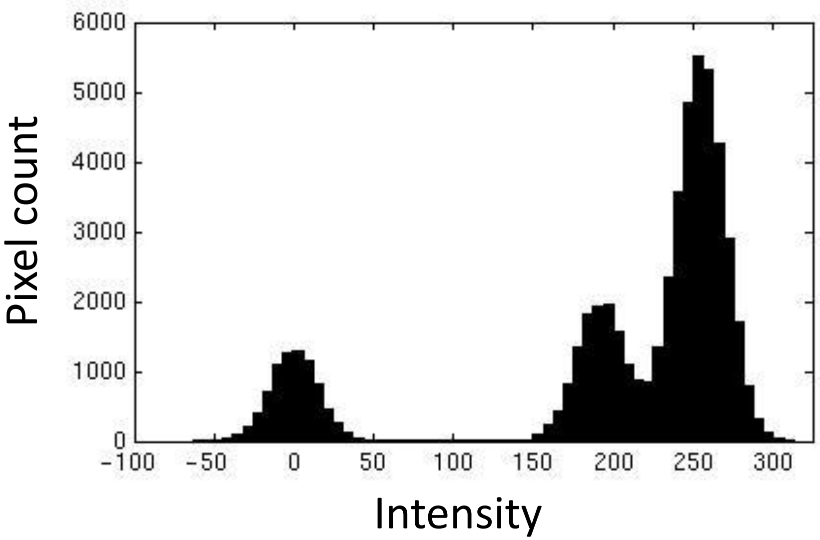


Slide credit: Kristen Grauman

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

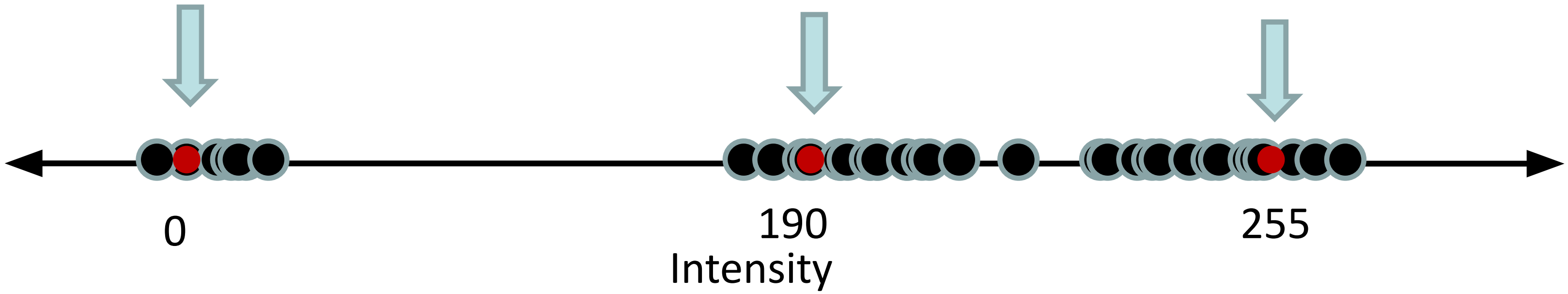


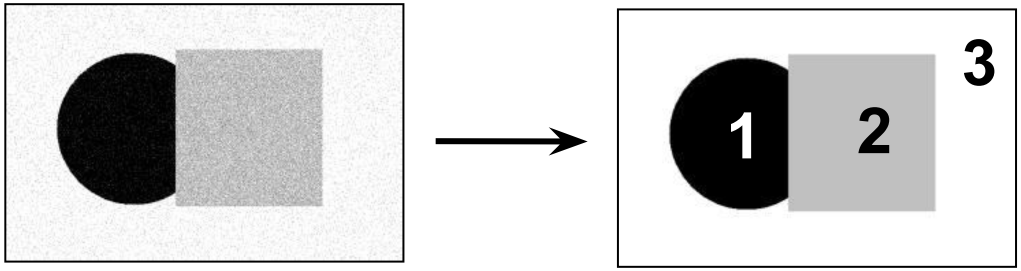
Input image



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

Slide credit: Kristen Grauman





- **General idea**: 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 defined as those that minimize Sum of Square Distance (SSD) between all points and their nearest cluster center c_i:

$$SSD = \sum_{cluster\,i} \sum_{x \in cluster\,i} (x - c_i)^2$$

Slide credit: Kristen Grauman

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Clustering for Summarization

Goal: cluster to minimize variance in data given clusters

– Preserve information

Cluster center Data

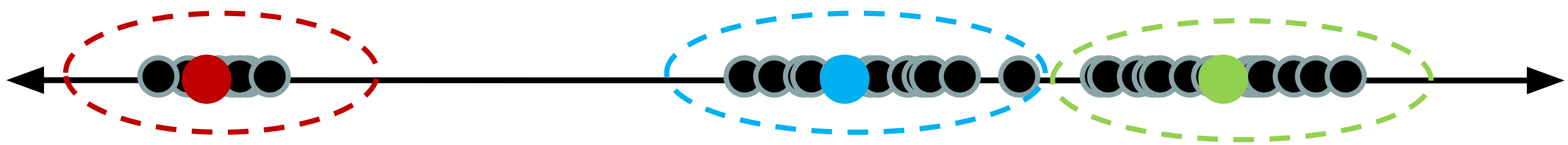
$$c^*,\ \delta^* = \underset{c,\,\delta}{\arg\min} \frac{1}{N} \sum_{j}^{N} \sum_{i}^{K} \delta_{ij} \left(c_i - x_j\right)^2$$

Whether x_j is assigned to c_i

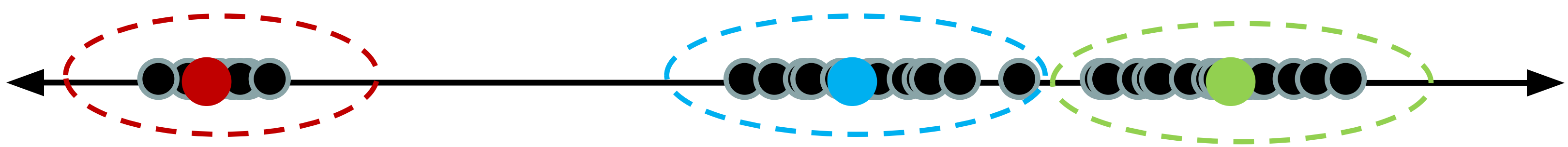
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Clustering - but how?

- 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: Kristen Grauman

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

K-means clustering

1. Initialize ($t = 0$): cluster centers $c_1, \ldots, c_K$

2. Compute δ^t: assign each point to the closest center
   – δ^t denotes the set of assignment for each x_j to cluster c_i at iteration *t*

$$\delta^t = \underset{\delta}{\operatorname{argmin}} \frac{1}{N} \sum_{j}^{N} \sum_{i}^{K} \delta_{ij}^{t-1} \left(c_i^{t-1} - x_j \right)^2$$

3. Computer c^t: update cluster centers as the mean of the points

$$c^t = \underset{c}{\operatorname{argmin}} \frac{1}{N} \sum_{j}^{N} \sum_{i}^{K} \delta_{ij}^{t} \left(c_i^{t-1} - x_j \right)^2$$

4. Update $t = t + 1$, Repeat Step 2-3 till stopped

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Slide: Derek Hoiem

K-means clustering

1. Initialize ($t = 0$): cluster centers $c_1, \ldots, c_K$
   - Commonly used: random initialization

2. Compute δ^t: assign each point to the closest center

3. Computer c^t: update cluster centers as the mean of the points

$$c^t = \underset{c}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij}^t \left(c_i^{t-1} - x_j \right)^2$$

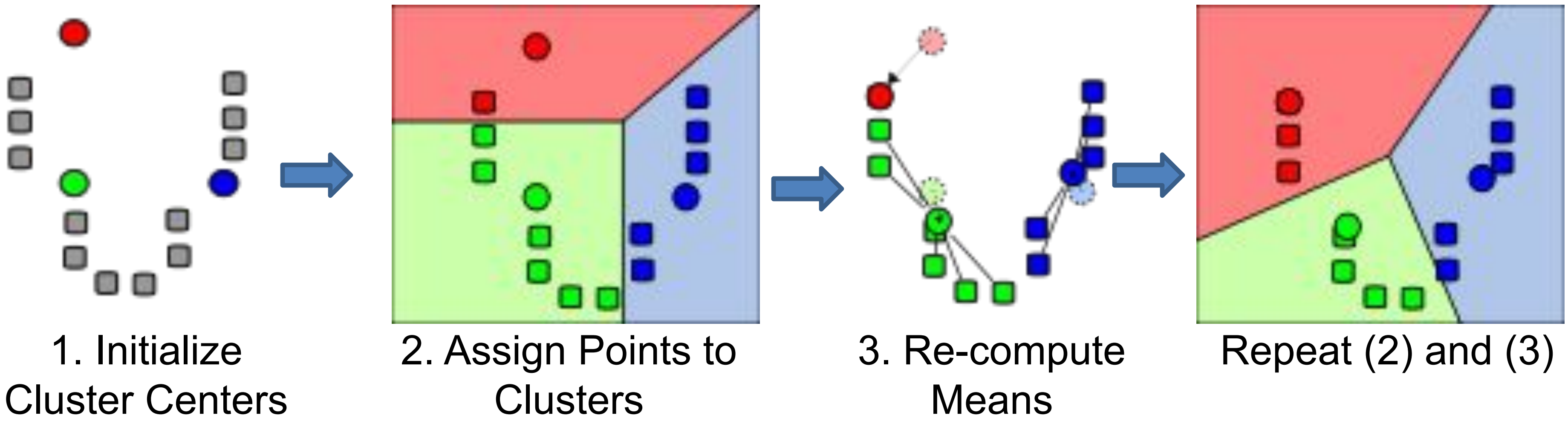
4. Update $t = t + 1$, Repeat Step 2-3 till stopped

   - c^t (or assignments) doesn't change anymore
     $\Rightarrow$ Convergence, stop.

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Slide: Derek Hoiem

K-means clustering



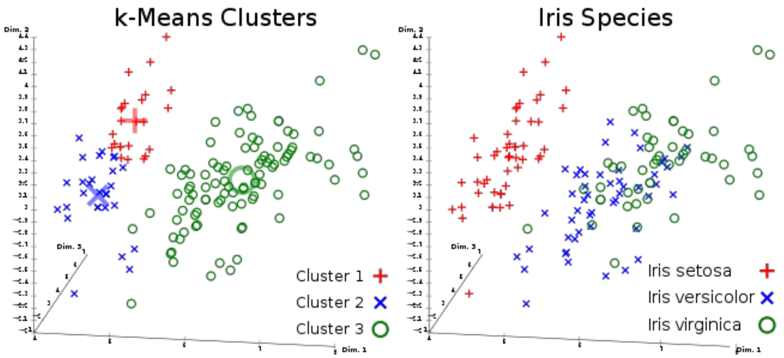
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Illustration Source: wikipedia

K-means clustering

- Converges to a *local minimum* solution
  - Initialize multiple runs

- Better fit for spherical data



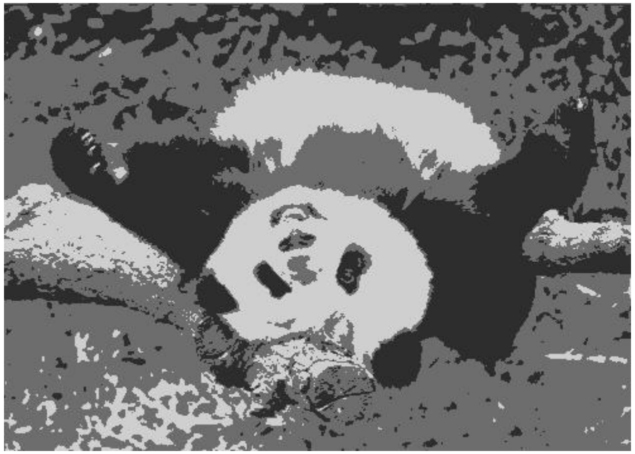
- Need to pick K (# of clusters)

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Segmentation as Clustering

Original image

2 clusters



3 clusters

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Feature Space

- Depending on what we choose as the *feature space (=representation)*, we can group pixels in different ways.

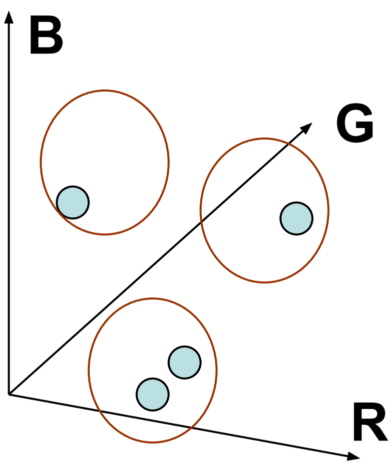
- Grouping pixels based on intensity similarity

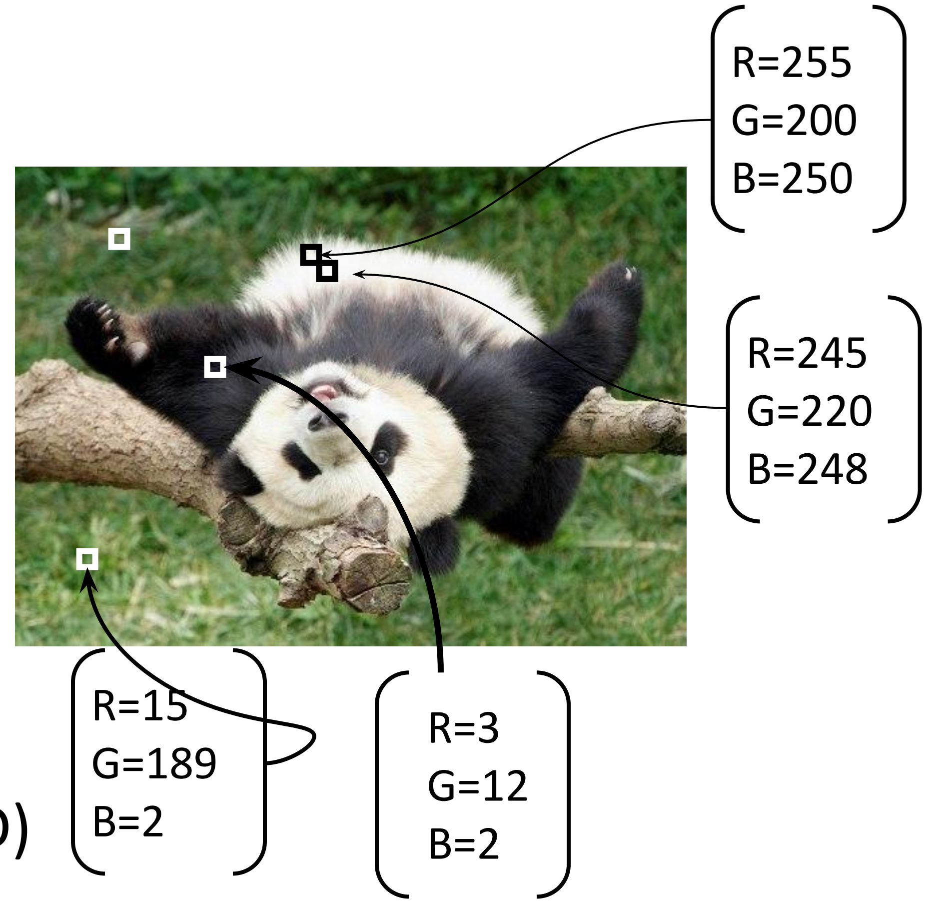
- Feature space: intensity value (1D)

Slide credit: Kristen Grauman

Feature Space

- Depending on what we choose as the *feature space*, we can group pixels in different ways.
- Grouping pixels based on color similarity





- Feature space: color value (3D)

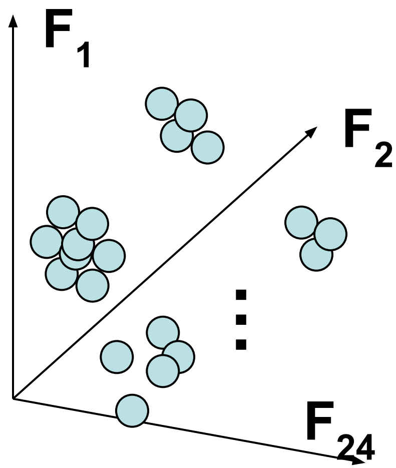
Slide credit: Kristen Grauman

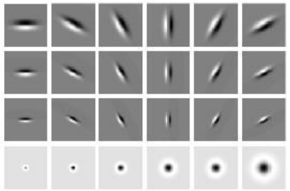
ebles, and Ranjay Krishna

Feature Space

- 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., 24D)

Slide credit: Kristen Grauman

Features greatly affect K-Means

Image | Intensity-based clusters | Color-based clusters

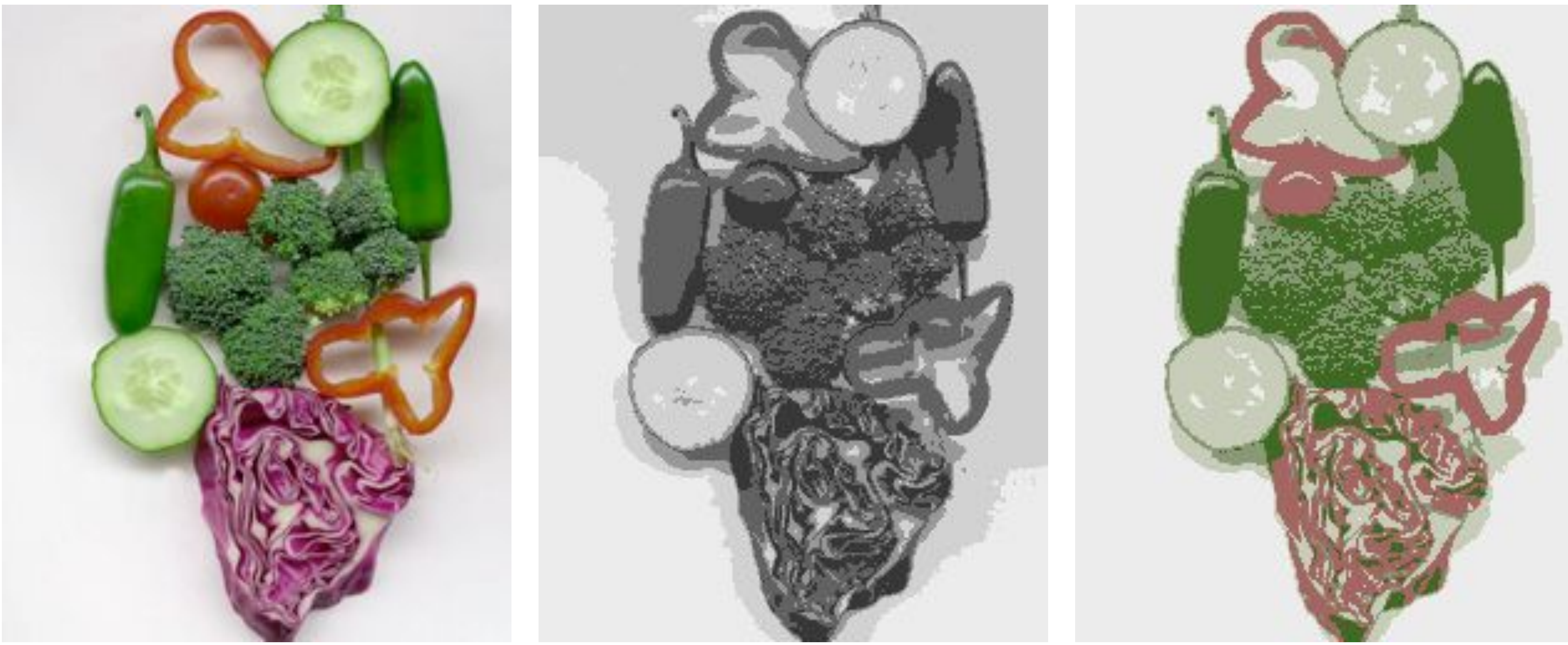
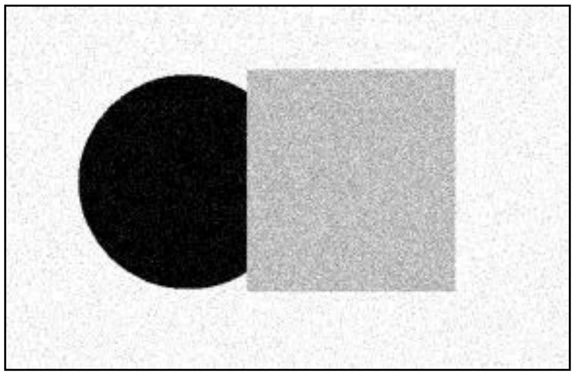


Image source: Forsyth & Ponce

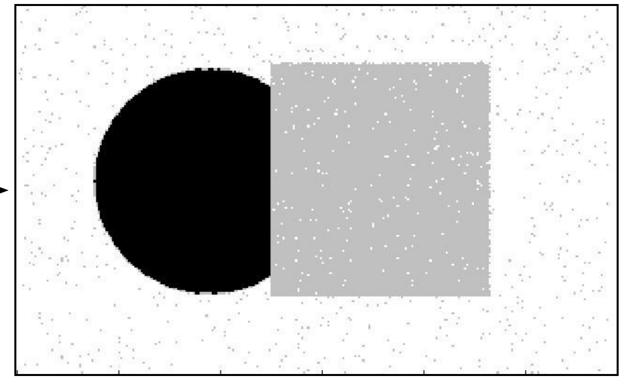
Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

Smoothing Out Cluster Assignments

- Assigning a cluster label per pixel may yield outliers:

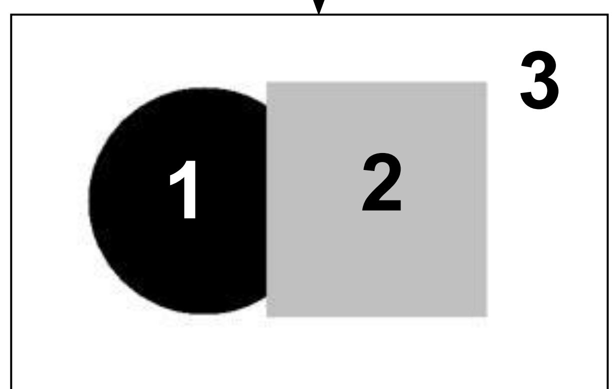


Original



Labeled by cluster center's intensity

?



- How can we ensure they are spatially smooth?

Slide credit: Kristen Grauman

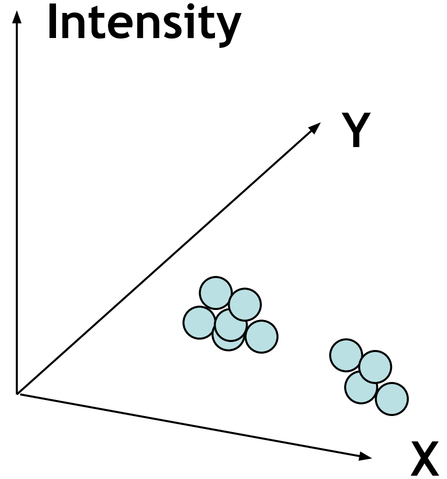
…ebles, and Ranjay Krishna

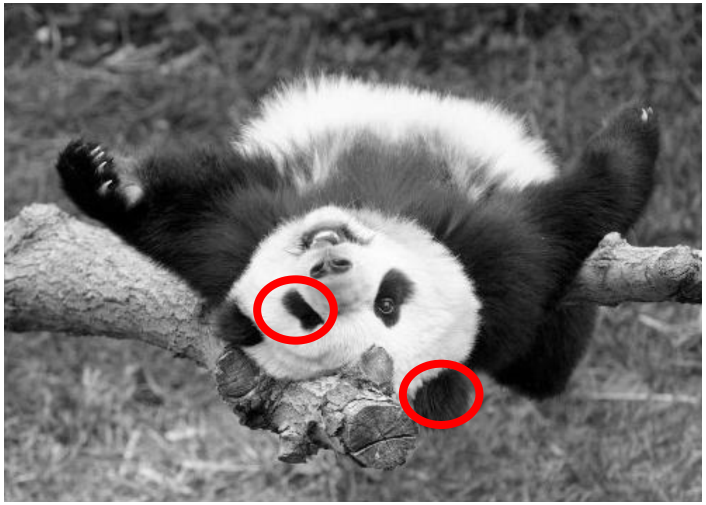
Segmentation as Clustering

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

- Grouping pixels based on *intensity+position* similarity

-





⇒ Way to encode both *similarity* and *proximity.*

Slide credit: Kristen Grauman

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

How to evaluate clusters?

- Generative
  - How well are points reconstructed from the clusters?
- Discriminative
  - How well do the clusters correspond to labels?
    - Can we correctly classify which pixels belong to the panda?
  - Note: unsupervised clustering does not aim to be discriminative as we don’t have the labels.

Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

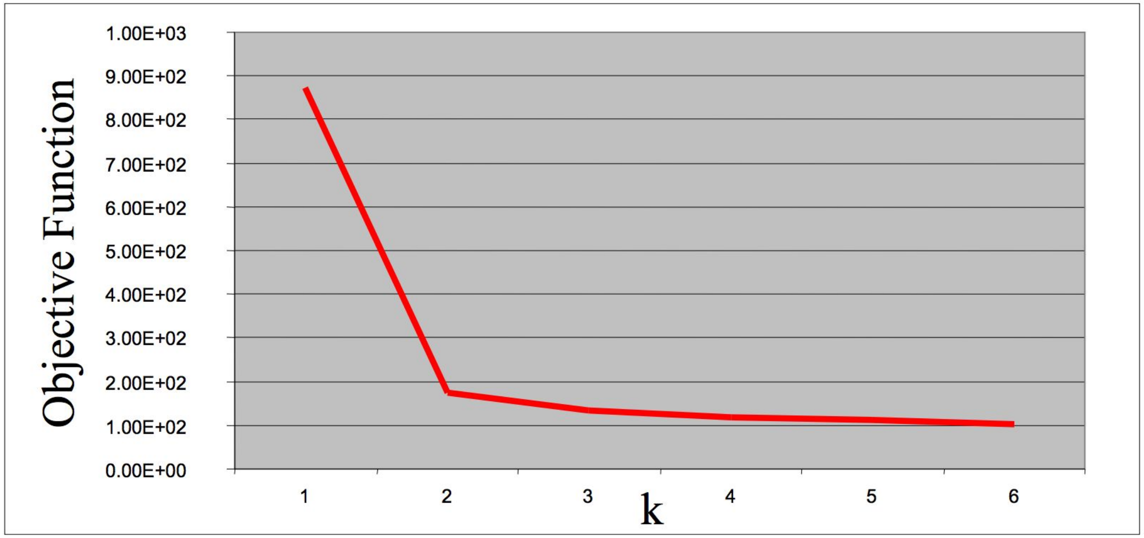
Slide: Derek Hoiem

How to choose the number of clusters?

Try different numbers of clusters in a validation set and look at performance.

We can plot the objective function values for k equals 1 to 6…

The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as “knee finding” or “elbow finding”.

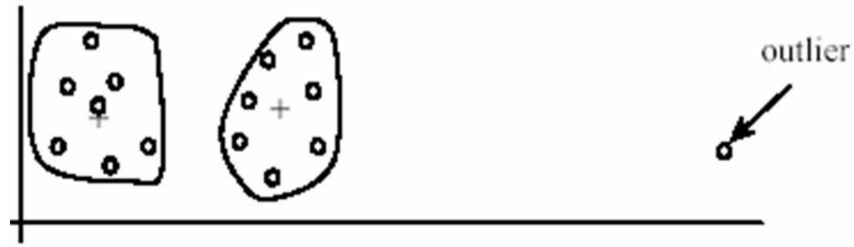


Adapted from slides by Juan Carlos Niebles, and Ranjay Krishna

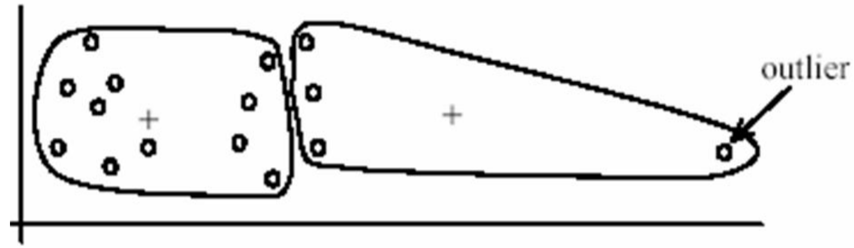
Slide: Derek Hoiem

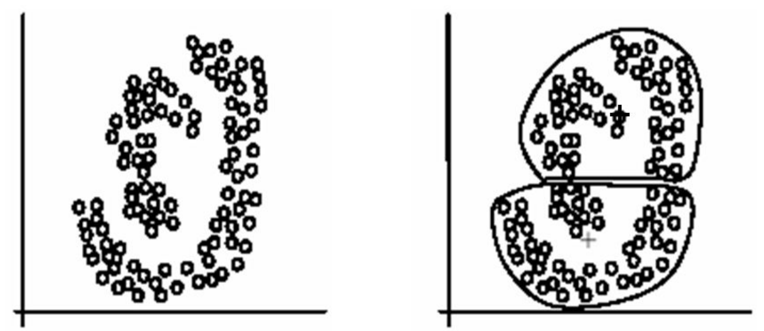
K-Means pros and cons

- Pros
  - Finds cluster centers that minimize conditional variance (good representation of data)
  - Simple and fast, Easy to implement
- Cons
  - Need to choose K
  - Sensitive to outliers
  - Prone to local minima
  - All clusters have the same parameters (e.g., distance measure is non-adaptive)
  - *Can be slow: each iteration is O(KNd) for N d-dimensional points
- Usage
  - Unsupervised clustering
  - Rarely used for pixel segmentation



(B): Ideal clusters





(A): Two natural clusters (B): *k*-means clusters