



Lecture: Segmentation

Juan Carlos Niebles and Ranjay Krishna
Stanford Vision and Learning Lab



CS 131 Roadmap





What we will learn today

- Introduction to segmentation and clustering
- Gestalt theory for perceptual grouping
- Agglomerative clustering
- Oversegmentation

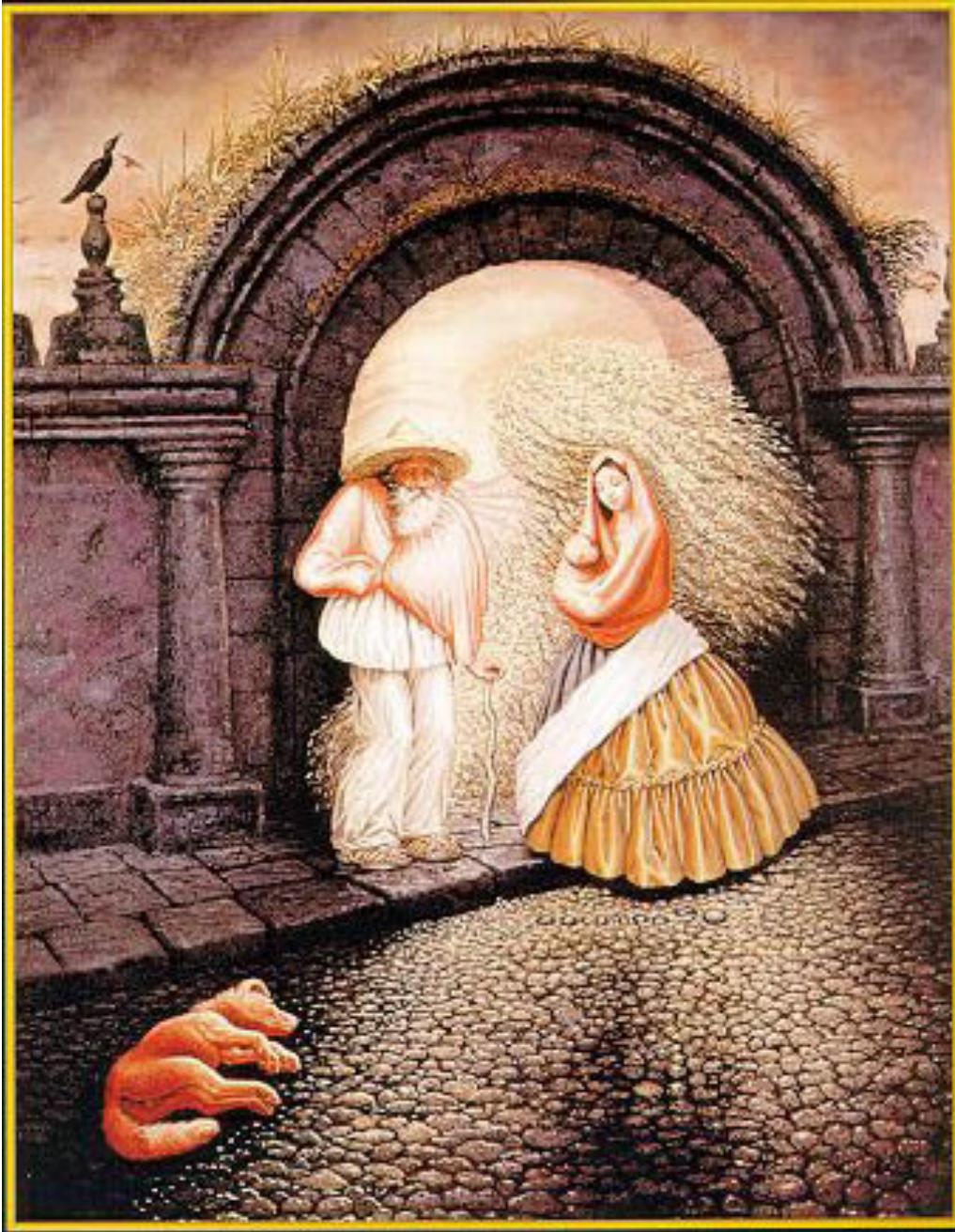
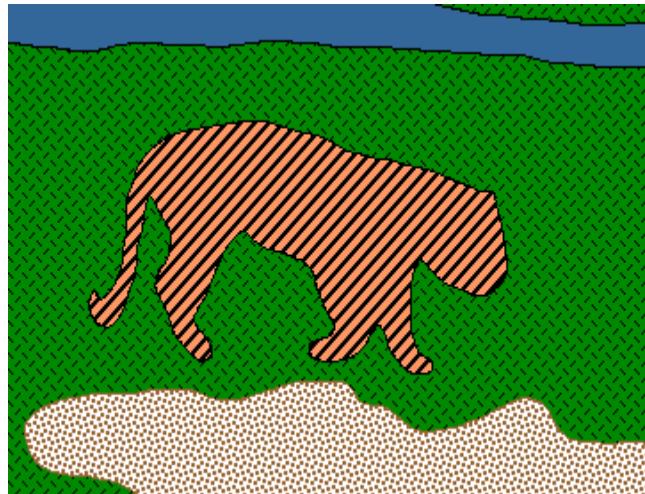




Image Segmentation

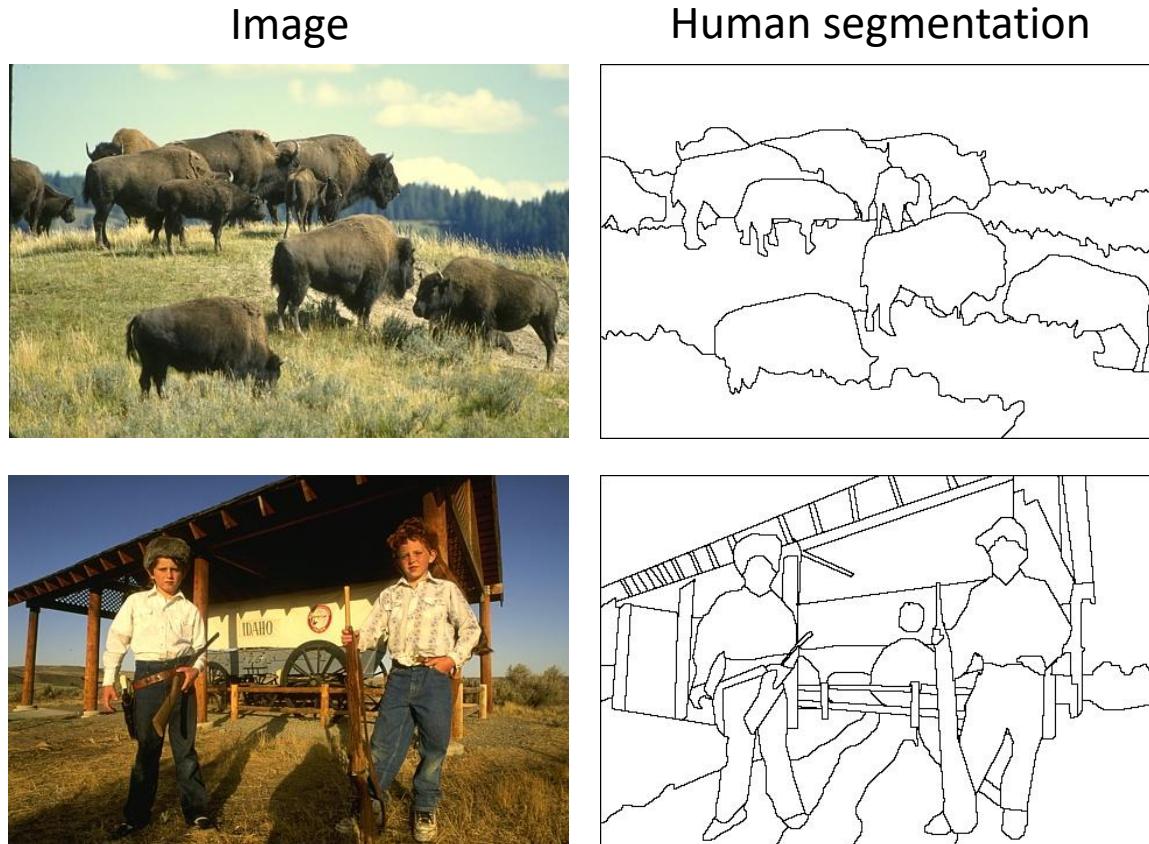
- Goal: identify groups of pixels that go together





The Goals of Segmentation

- Separate image into coherent “objects”

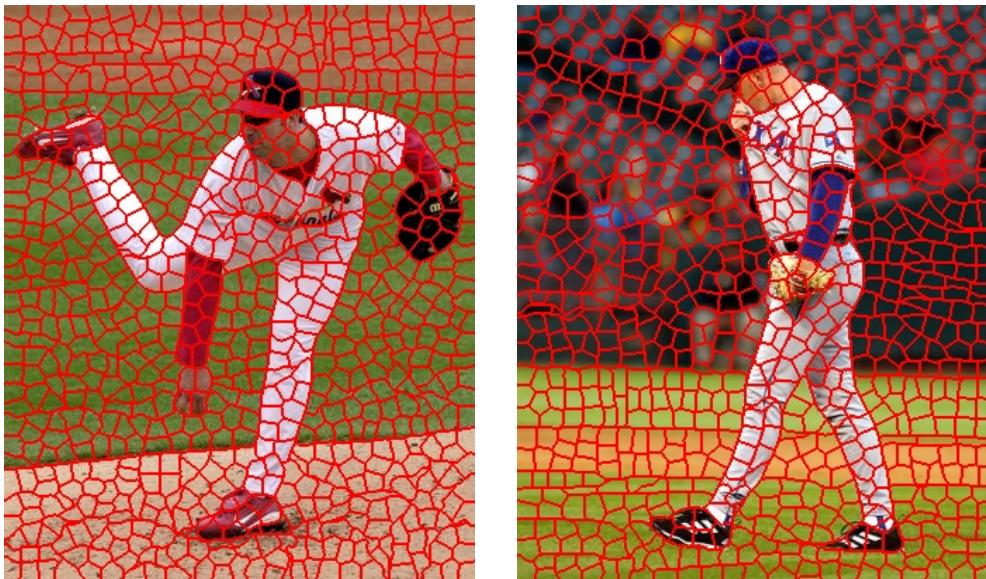




The Goals of Segmentation

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

“superpixels”



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

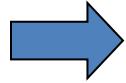


Segmentation for feature support

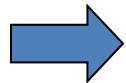




Segmentation for efficiency



[Felzenszwalb and Huttenlocher 2004]



[Shi and Malik 2001]

[Hoiem et al. 2005, Mori 2005]



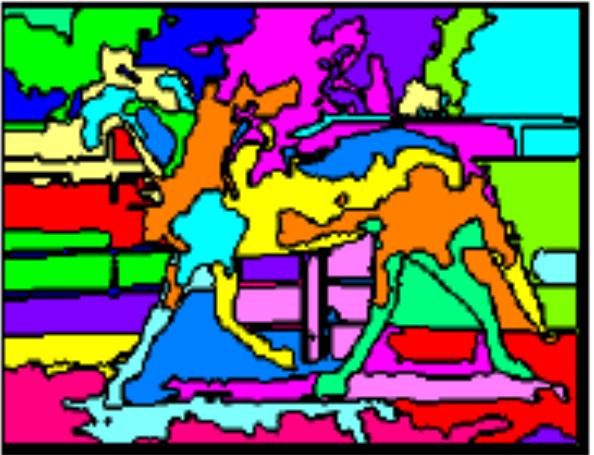
Segmentation as a result



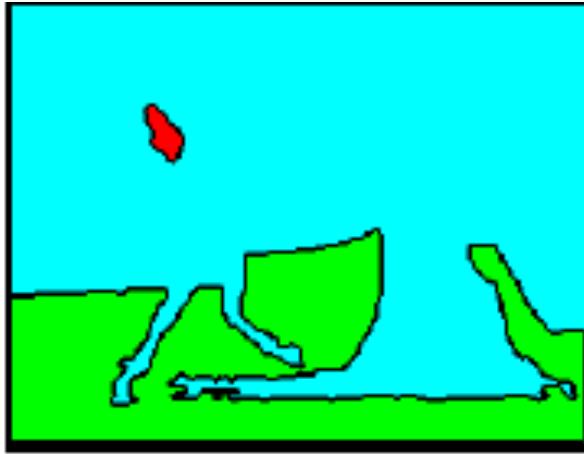
Rother et al. 2004



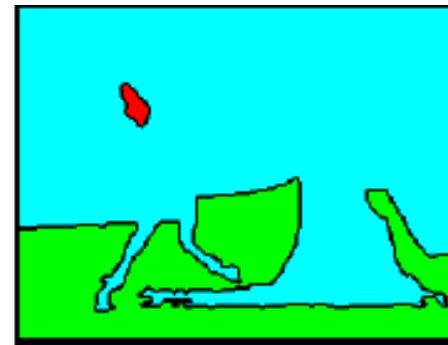
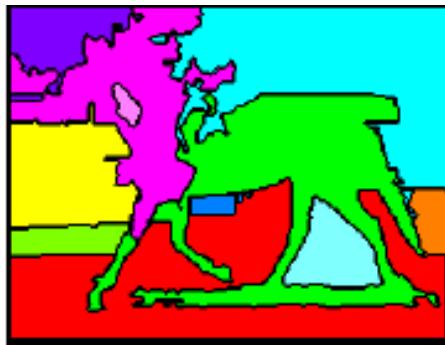
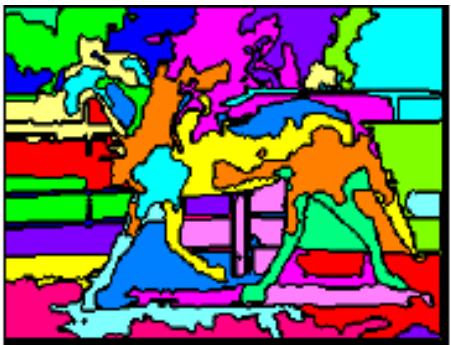
Types of segmentations



Oversegmentation



Undersegmentation



Multiple Segmentations



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?



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



How do we cluster?

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

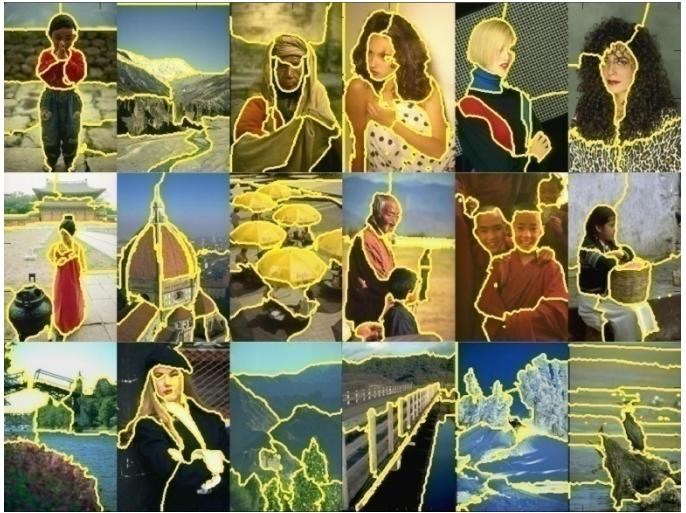


General ideas

- **Tokens**
 - whatever we need to group (pixels, points, surface elements, etc., 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...)
- > These two are not mutually exclusive



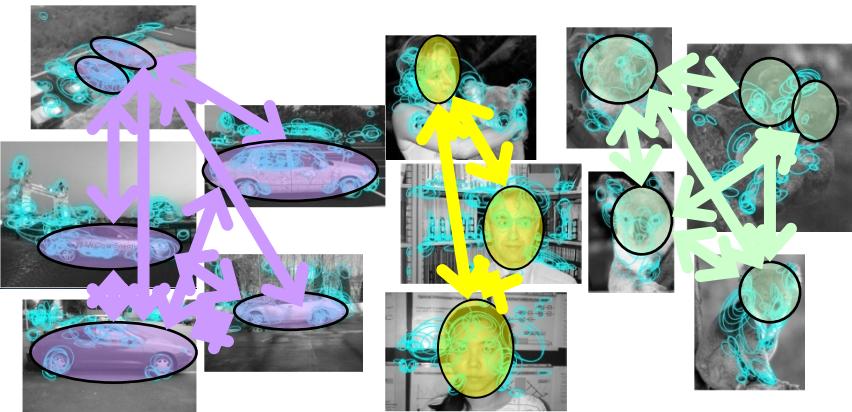
Examples of Grouping in Vision



Determining image regions

*What things should
be grouped?*

*What cues
indicate groups?*



Object-level grouping



Grouping video frames into shots



Figure-ground



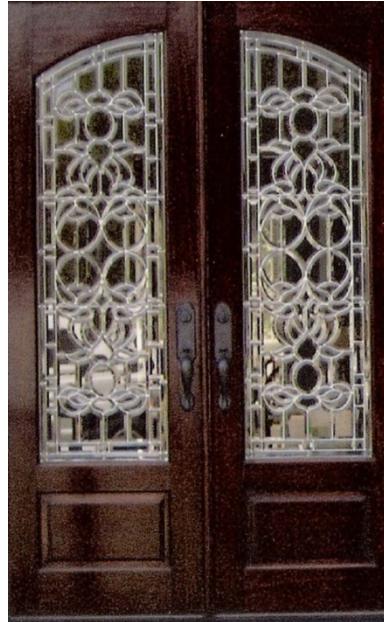
Similarity



Slide credit: Kristen Grauman



Symmetry



Slide credit: Kristen Grauman



Common Fate



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



(c) 2005 Heiko Burkhardt, ilm0.00m



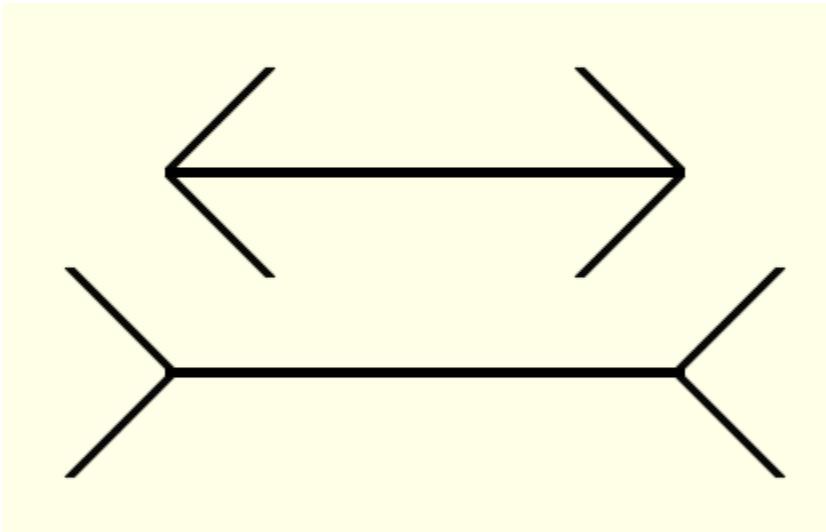
Proximity



Slide credit: Kristen Grauman



Muller-Lyer Illusion



- What makes the bottom line look longer than the top line?



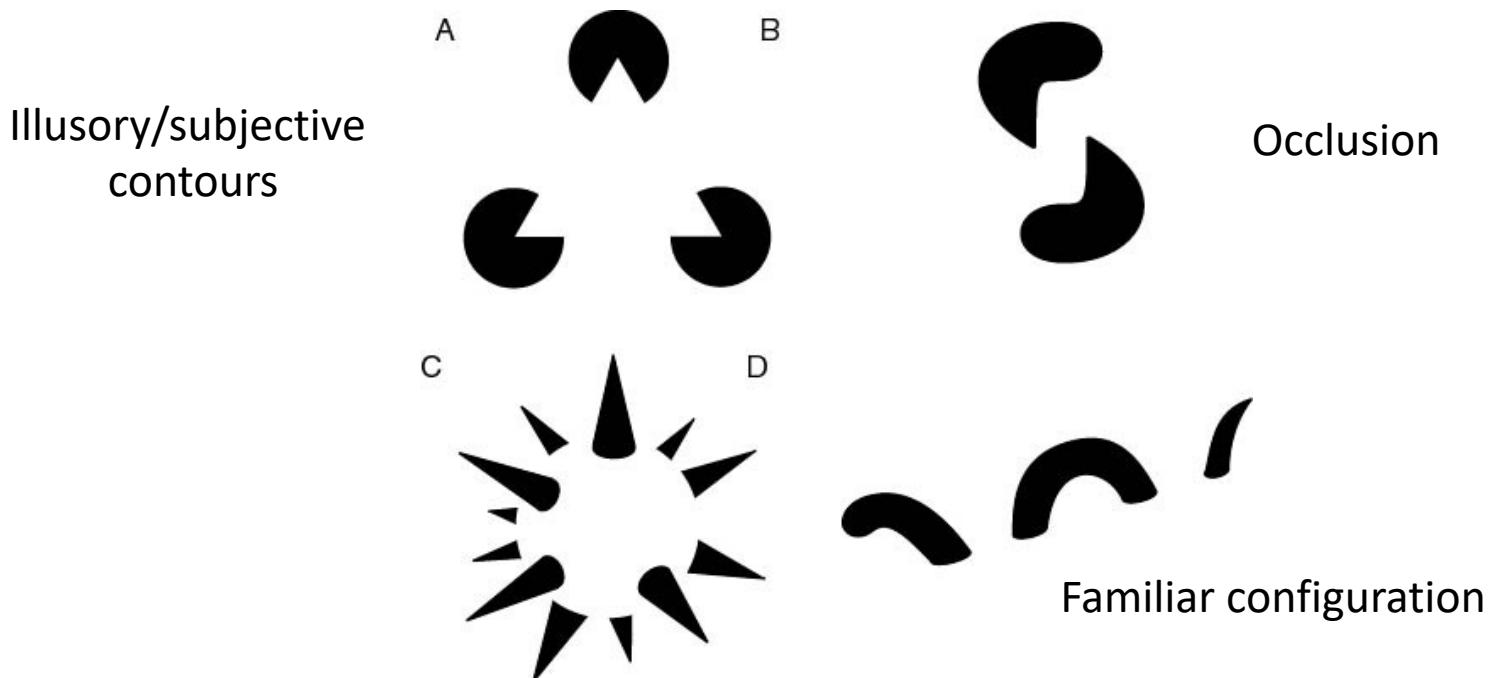
What we will learn today

- Introduction to segmentation and clustering
- Gestalt theory for perceptual grouping
- Agglomerative clustering
- Oversegmentation



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”



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

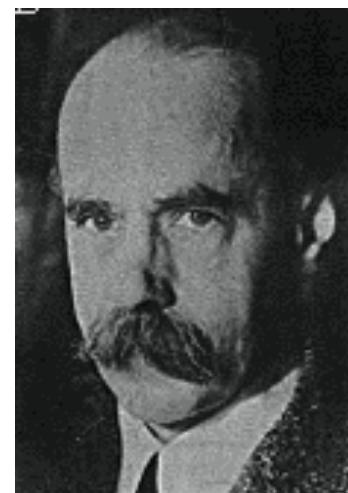


Gestalt Theory

- 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>



Gestalt Factors



Not grouped



Proximity



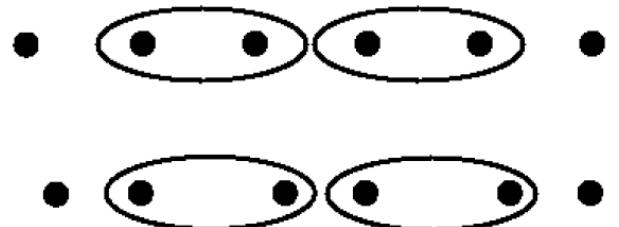
Similarity



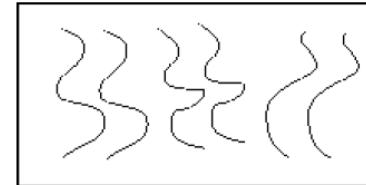
Similarity



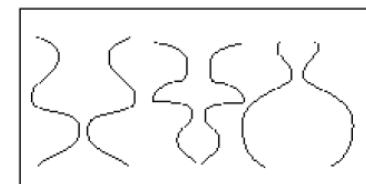
Common Fate



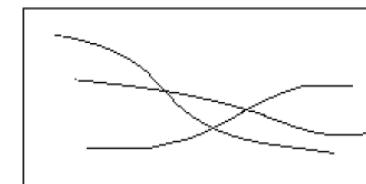
Common Region



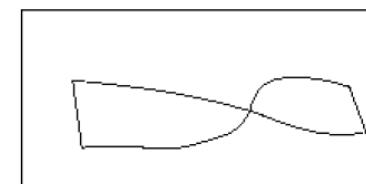
Parallelism



Symmetry



Continuity

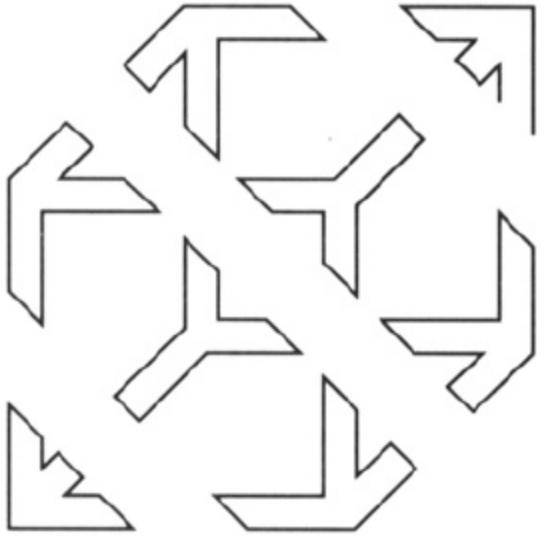


Closure

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

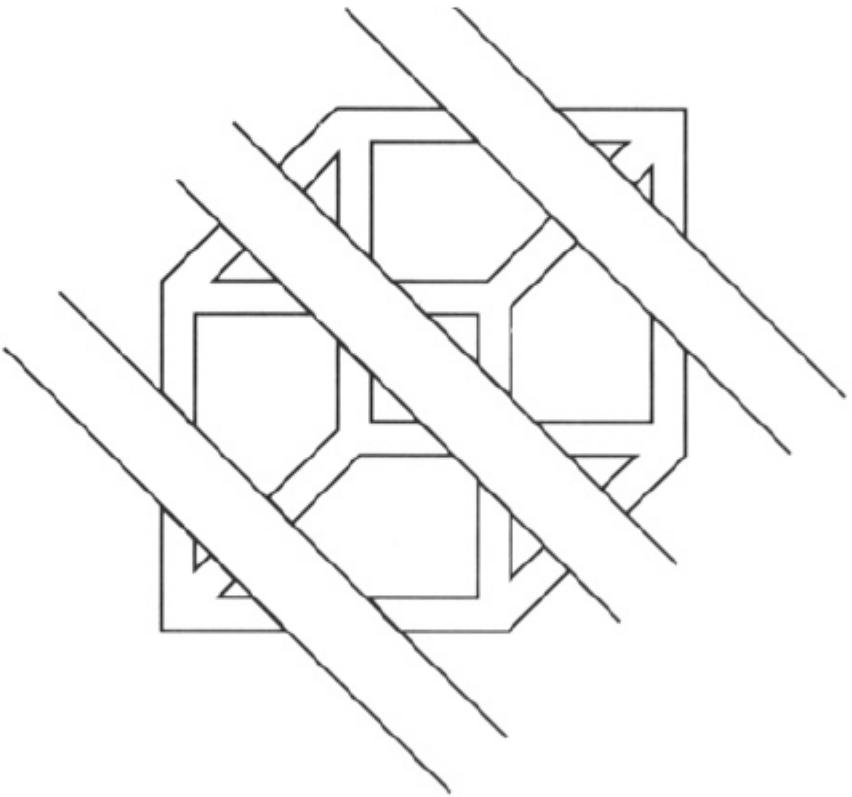


Continuity through Occlusion Cues





Continuity through Occlusion Cues



Continuity, explanation by occlusion



Continuity through Occlusion Cues

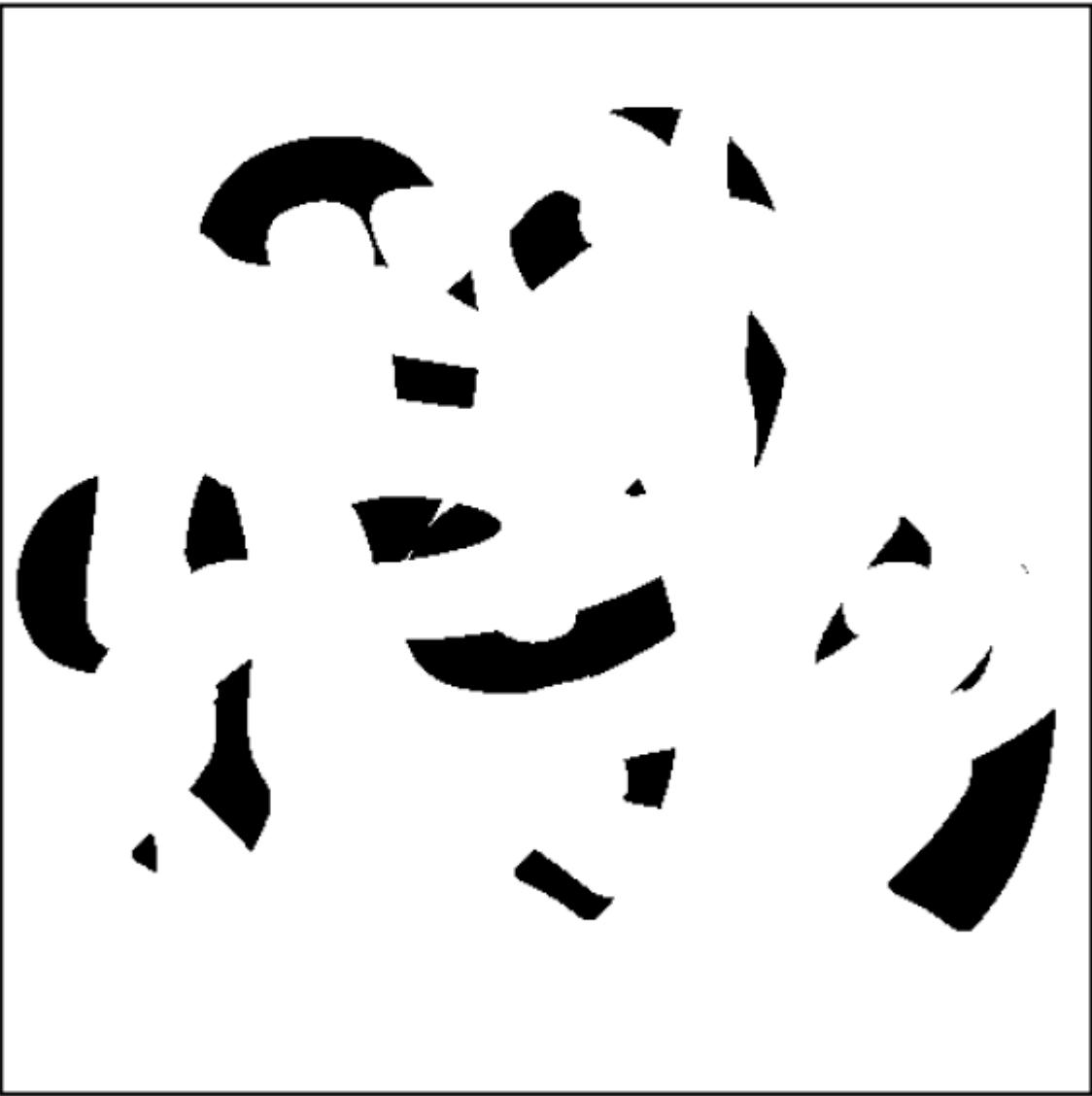


Image source: Forsyth & Ponce



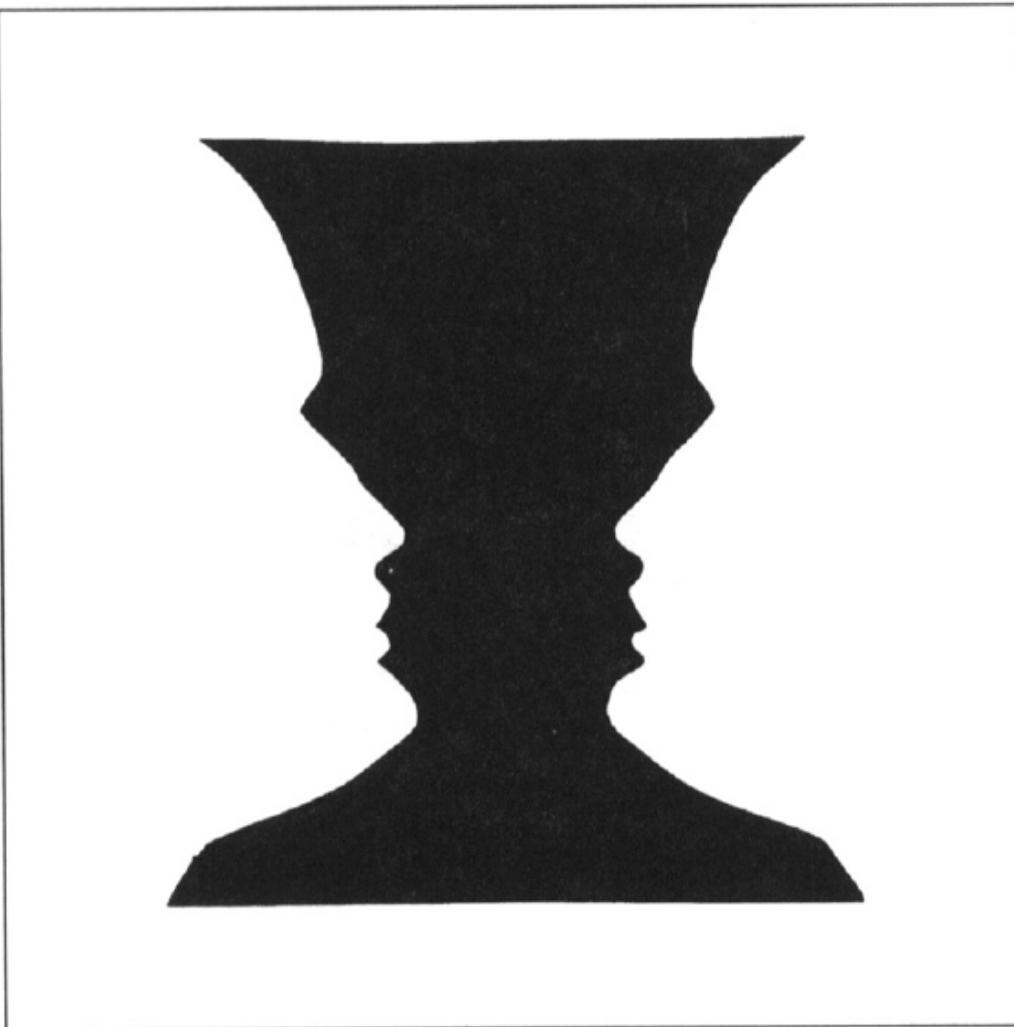
Continuity through Occlusion Cues



Image source: Forsyth & Ponce



Figure-Ground Discrimination





The Ultimate Gestalt?





What we will learn today

- Introduction to segmentation and clustering
- Gestalt theory for perceptual grouping
- **Agglomerative clustering**
- Oversegmentation



Clustering: distance measure

Clustering is an unsupervised learning method. Given items $x_1, \dots, x_n \in \mathbb{R}^D$, the goal is to group them into clusters. We need a pairwise distance/similarity function between items, and sometimes the desired number of clusters.

When data (e.g. images, objects, documents) are represented by feature vectors, commonly used measures are:

- *Euclidean distance.*
- *cosine similarity.*



Defining Distance Measures

Let x and x' be two objects from the universe of possible objects.

The distance (similarity) between x and x' is a real number denoted by $\text{sim}(x, x')$.

- The Euclidean distance is defined as

$$\text{dist}(x, x') = \sqrt{\sum (x_i - x'_i)^2}$$

- In contrast, the cosine similarity measure would be

$$\begin{aligned}\text{sim}(x, x') &= \cos(\theta) \\ &= \frac{x^\top x'}{\|x\| \cdot \|x'\|} \\ &= \frac{x^\top x'}{\sqrt{x^\top x} \sqrt{x'^\top x'}}.\end{aligned}$$

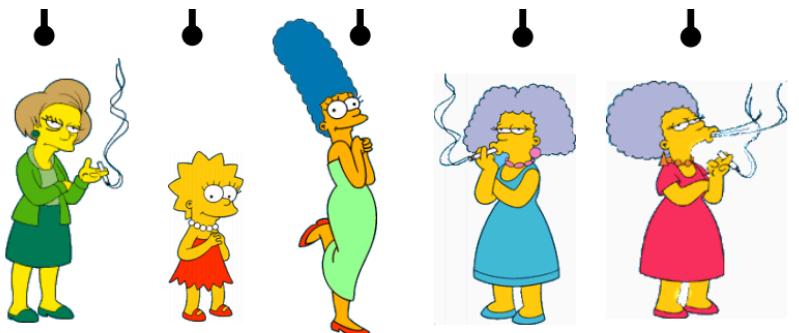


Desirable Properties of a Clustering Algorithms

- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine input parameters
- Interpretability and usability Optional
 - Incorporation of user-specified constraints



Animated example

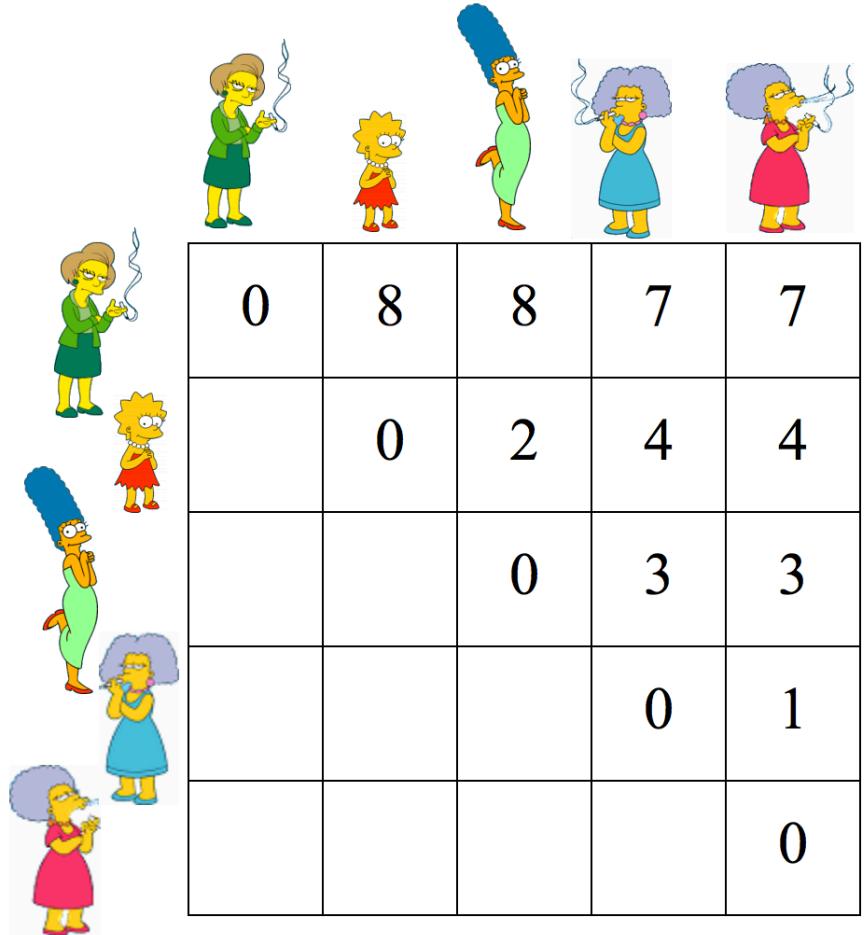
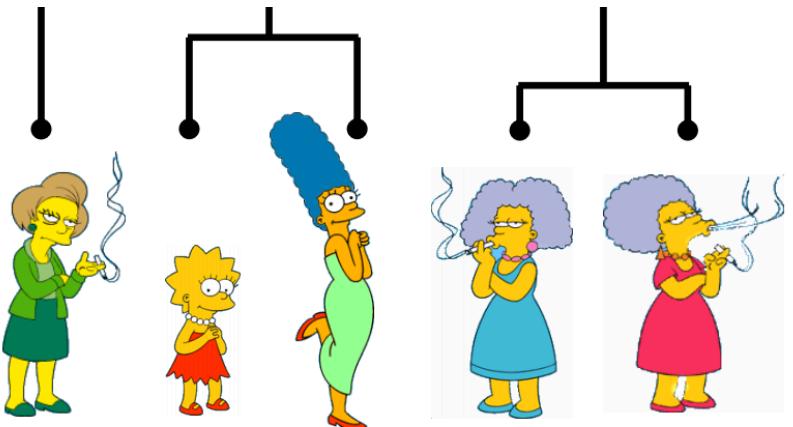


0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

[source](#)



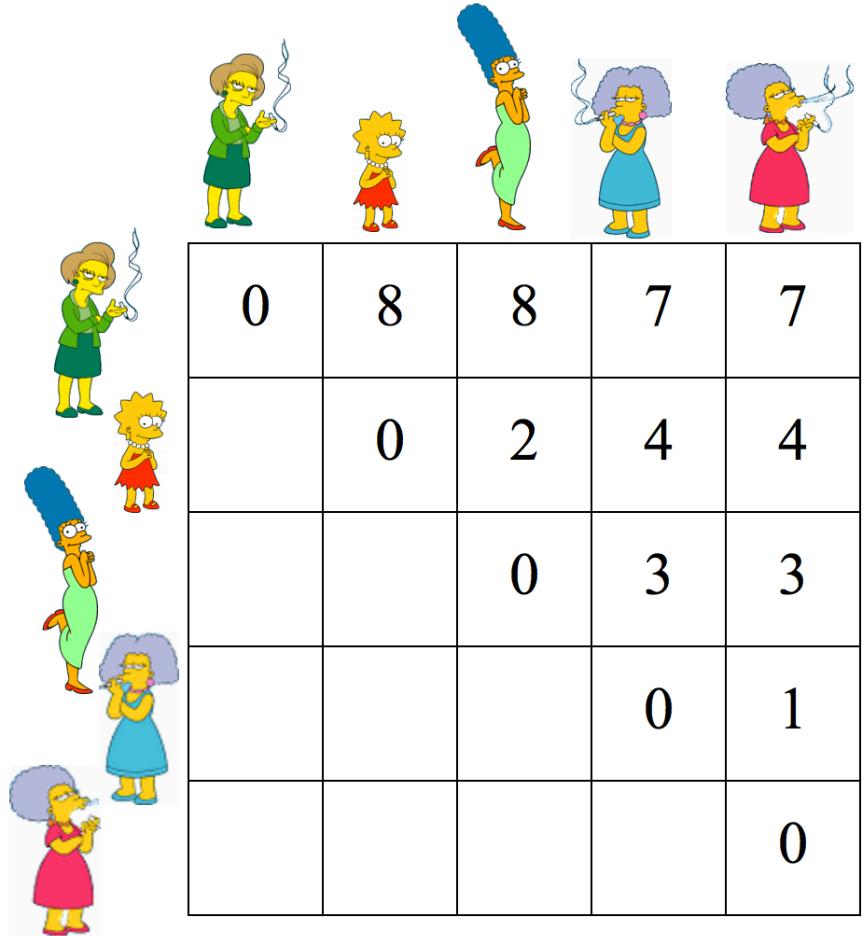
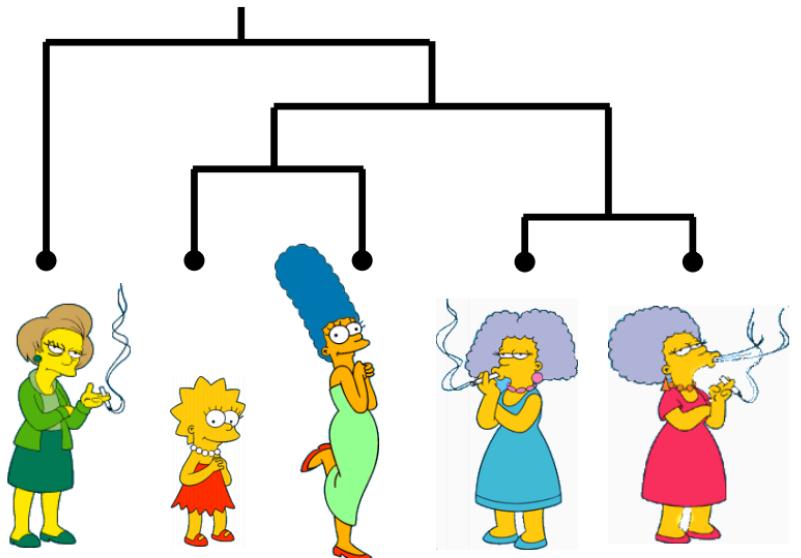
Animated example



[source](#)



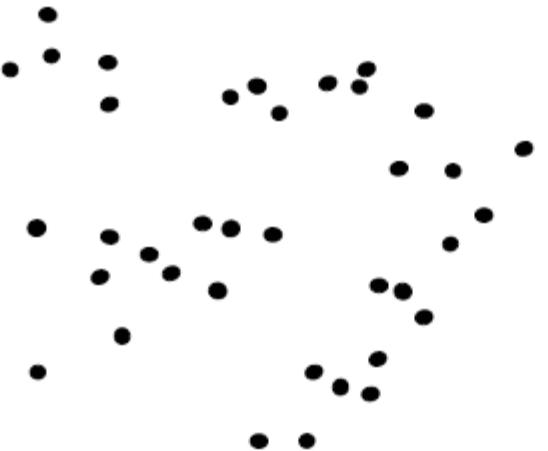
Animated example



[source](#)



Agglomerative clustering

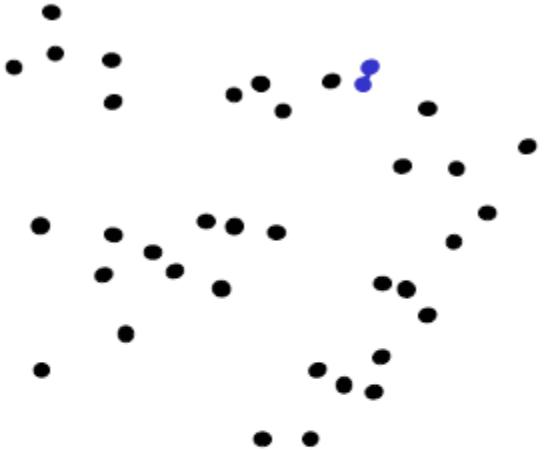


1. Say "Every point is its own cluster"

Slide credit: Andrew Moore



Agglomerative clustering



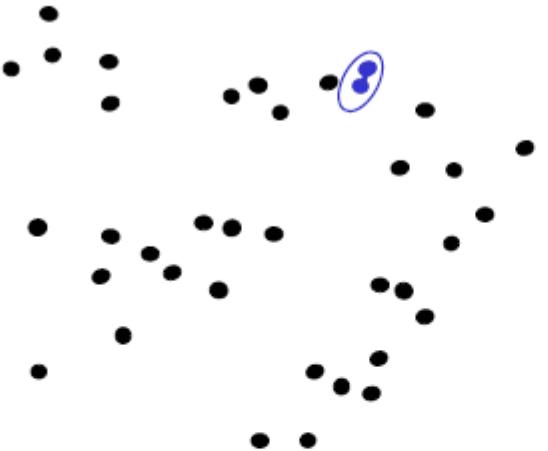
1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters



Slide credit: Andrew Moore



Agglomerative clustering



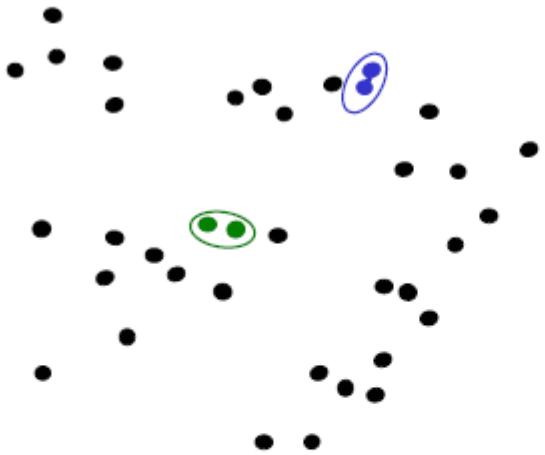
1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster



Slide credit: Andrew Moore



Agglomerative clustering



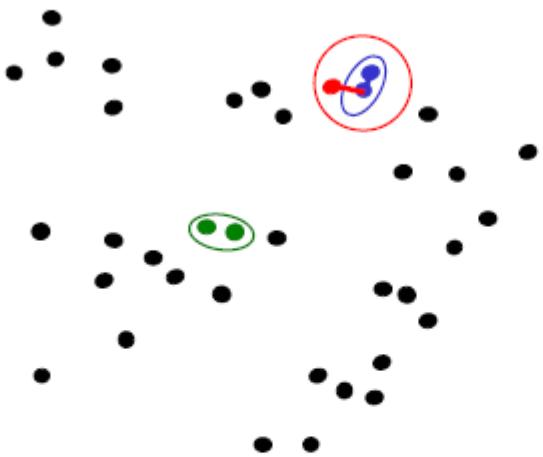
1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat



Slide credit: Andrew Moore



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



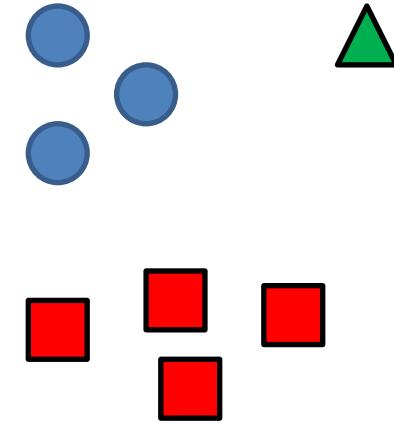
Slide credit: Andrew Moore



Agglomerative clustering

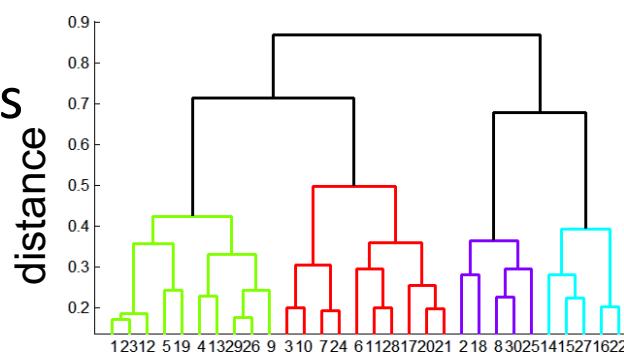
How to define cluster similarity?

- 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





Agglomerative Hierarchical Clustering - Algorithm

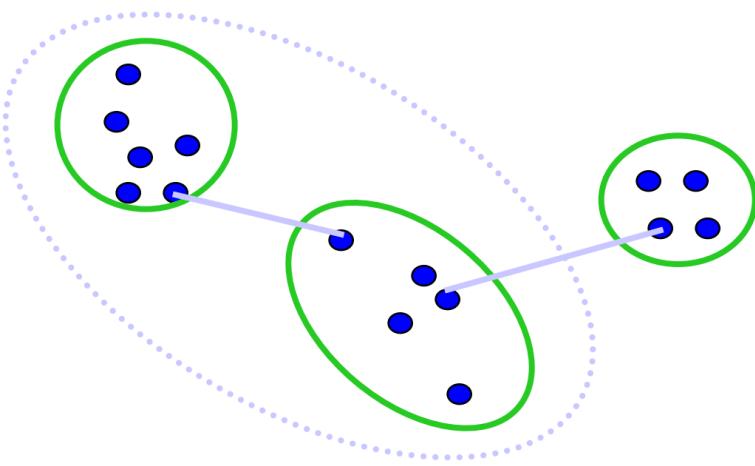
1. Initially each item x_1, \dots, x_n is in its own cluster C_1, \dots, C_n .
2. Repeat until there is only one cluster left:
 3. Merge the nearest clusters, say C_i and C_j .



Different measures of nearest clusters

Single Link

- $d(C_i, C_j) = \min_{x \in C_i, x' \in C_j} d(x, x')$. This is known as *single-linkage*. It is equivalent to the minimum spanning tree algorithm. One can set a threshold and stop clustering once the distance between clusters is above the threshold. Single-linkage tends to produce long and skinny clusters.



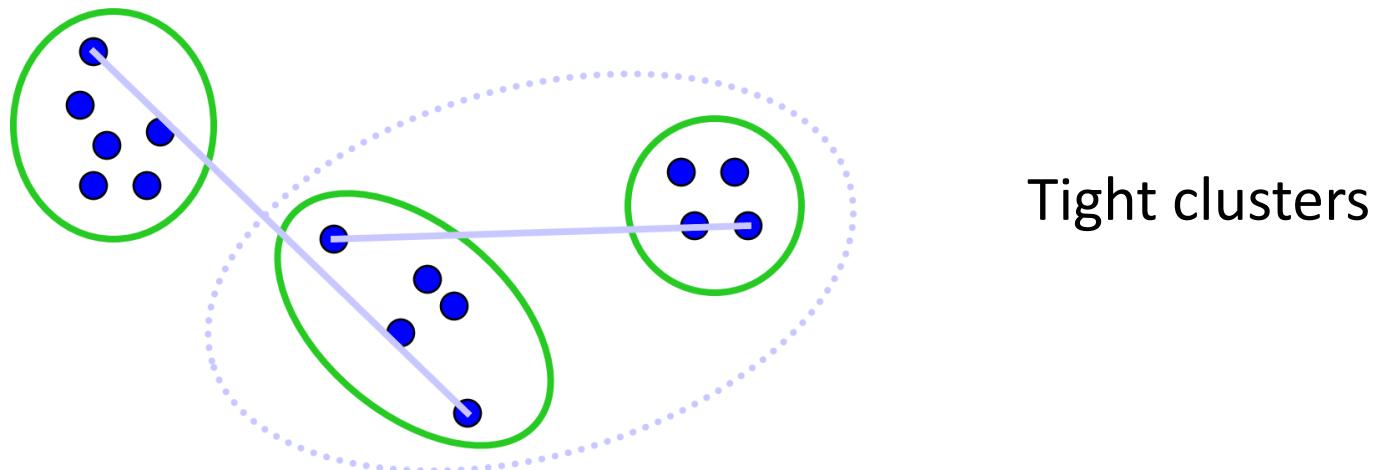
Long, skinny clusters



Different measures of nearest clusters

Complete Link

- $d(C_i, C_j) = \max_{x \in C_i, x' \in C_j} d(x, x')$. This is known as *complete-linkage*. Clusters tend to be compact and roughly equal in diameter.

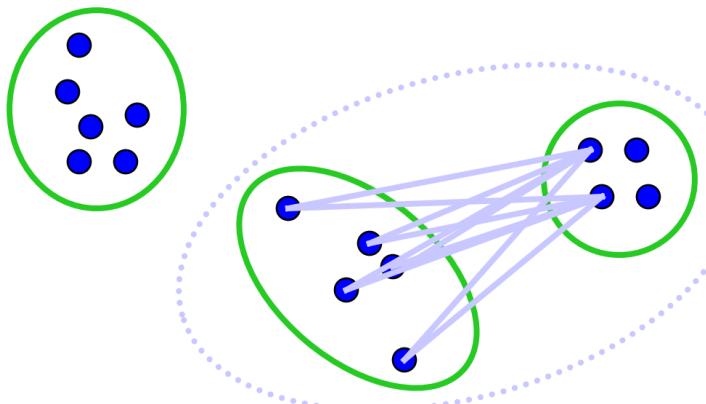




Different measures of nearest clusters

Average Link

- $d(C_i, C_j) = \frac{\sum_{x \in C_i, x' \in C_j} d(x, x')}{|C_i| \cdot |C_j|}$. This is the average distance between items. Somewhere between single-linkage and complete-linkage.



Robust against noise.



Conclusions: Agglomerative Clustering

Good

- Simple to implement, widespread application.
- Clusters have adaptive shapes.
- Provides a hierarchy of clusters.
- No need to specify number of clusters in advance.

Bad

- May have imbalanced clusters.
- Still have to choose number of clusters or threshold.
- Does not scale well. Runtime of $O(n^3)$.
- Can get stuck at a local optima.



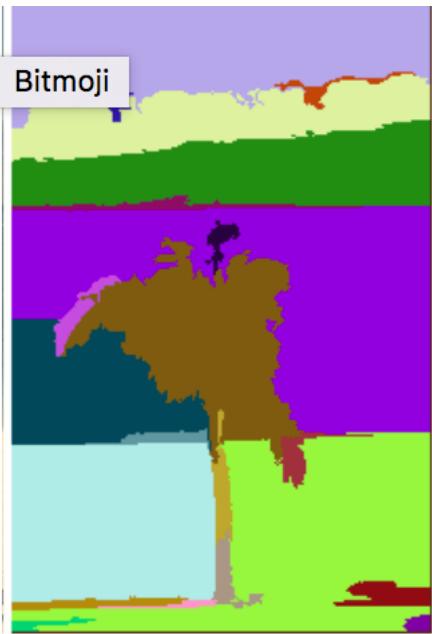
What we will learn today?

- Introduction to segmentation and clustering
- Gestalt theory for perceptual grouping
- Agglomerative clustering
- Oversegmentation



Over-segmenting images

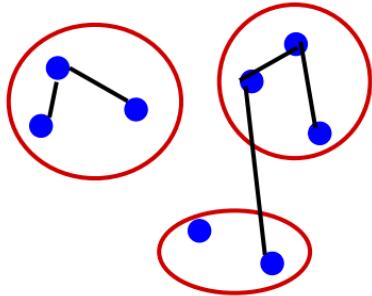
- Graph-based clustering for Image Segmentation
 - Introduced by *Felzenszwalb and Huttenlocher* in the paper titled *Efficient Graph-Based Image Segmentation*.





Problem Formulation

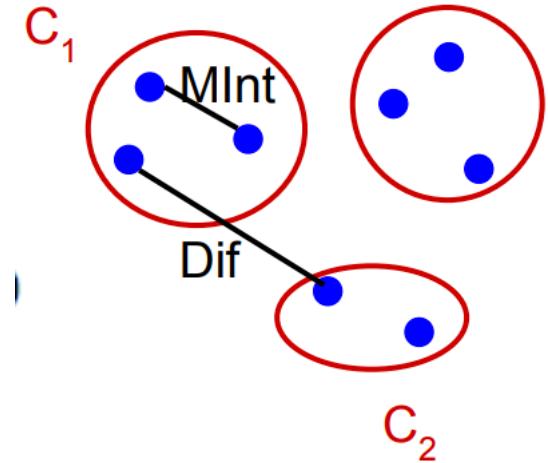
- Graph $G = (V, E)$
- V is set of nodes (i.e. pixels)
- E is a set of undirected edges between pairs of pixels
- $w(v_i, v_j)$ is the weight of the edge between nodes v_i and v_j .
- S is a segmentation of a graph G such that $G' = (V, E')$ where $E' \subset E$.
- S divides G into G' such that it contains distinct clusters C .





Predicate for segmentation

- Predicate D determines whether there is a boundary for segmentation.



$$\text{Merge}(C_1, C_2) = \begin{cases} \text{True} & \text{if } \text{dif}(C_1, C_2) < \text{in}(C_1, C_2) \\ \text{False} & \text{otherwise} \end{cases}$$

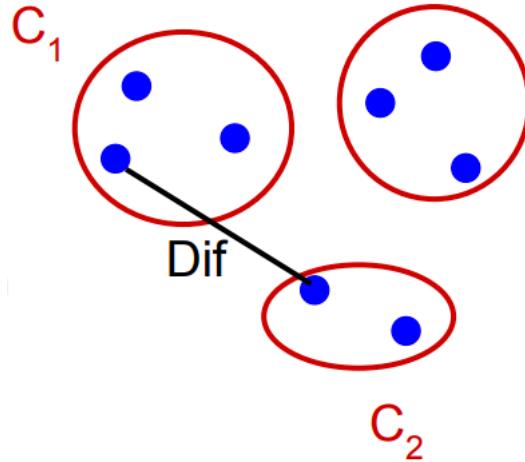
Where

- $\text{dif}(C_1, C_2)$ is the difference between two clusters.
- $\text{in}(C_1, C_2)$ is the internal difference in the clusters C_1 and C_2



Predicate for Segmentation

- Predicate D determines whether there is a boundary for segmentation.



$$\text{Merge}(C_1, C_2) = \begin{cases} \text{True} & \text{if } \text{dif}(C_1, C_2) < \text{in}(C_1, C_2) \\ \text{False} & \text{otherwise} \end{cases}$$

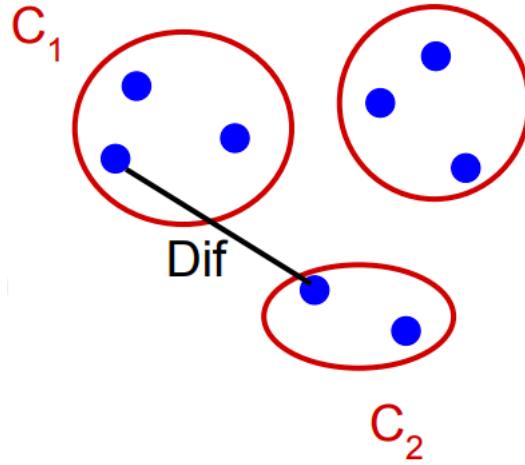
$$\text{dif}(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (C_1, C_2) \in E} w(v_i, v_j)$$

The difference between two components is the minimum weight edge that connects a node v_i in clusters C_1 to node v_j in C_2



Predicate for Segmentation

- Predicate D determines whether there is a boundary for segmentation.



$$\text{Merge}(C_1, C_2) = \begin{cases} \text{True} & \text{if } \text{dif}(C_1, C_2) < \text{in}(C_1, C_2) \\ \text{False} & \text{otherwise} \end{cases}$$

$$\text{dif}(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (C_1, C_2) \in E} w(v_i, v_j)$$

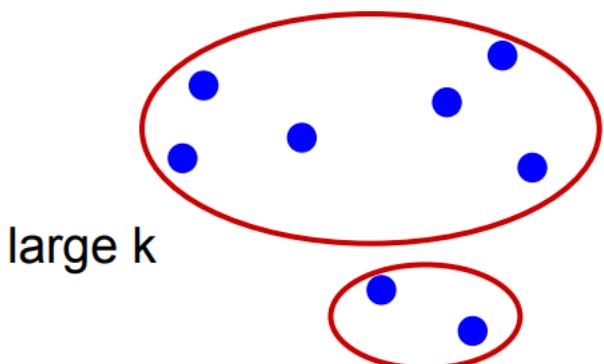
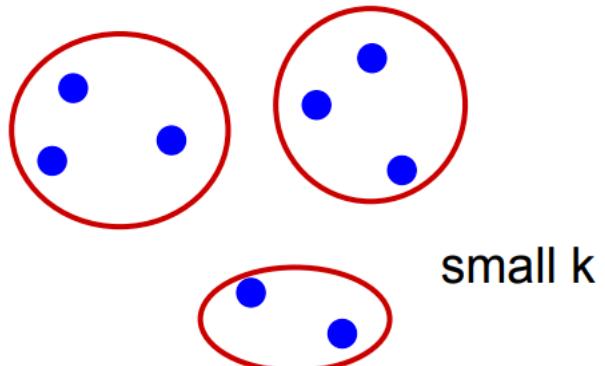
$$\text{in}(C_1, C_2) = \min_{C \in \{C_1, C_2\}} \left[\max_{v_i, v_j \in C} \left[w(v_i, v_j) + \frac{k}{|C|} \right] \right]$$

$\text{In}(C_1, C_2)$ is to the maximum weight edge that connects two nodes in the same component.



Predicate for Segmentation

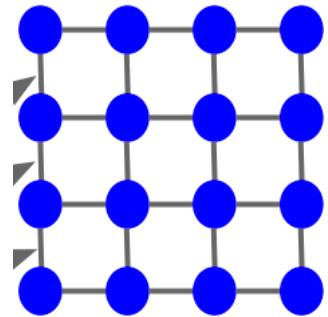
- $k/|C|$ sets the threshold by which the components need to be different from the internal nodes in a component.
- Properties of constant k :
 - If k is large, it causes a preference of larger objects.
 - k does not set a minimum size for components.





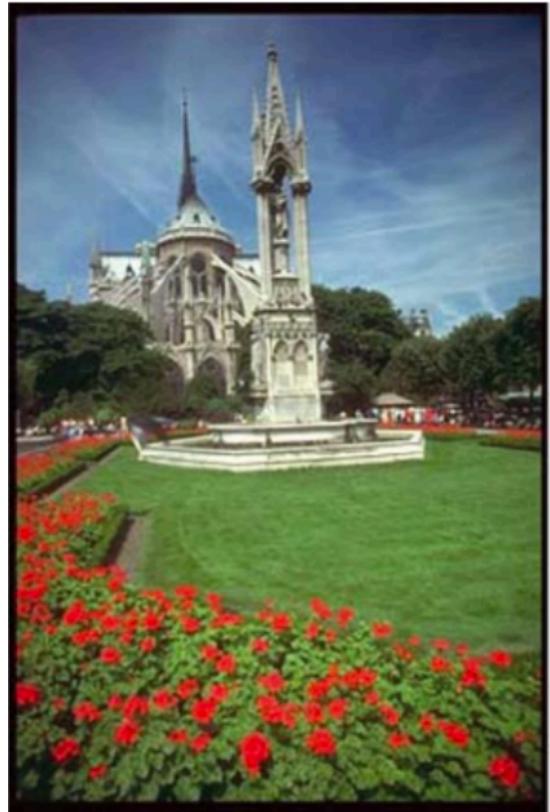
Features and weights

- Project every pixel into feature space defined by (x, y, r, g, b) .
- Every pixel is connected to its 8 neighboring pixels and the weights are determined by the difference in intensities.
- Weights between pixels are determined using L2 (Euclidian) distance in feature space.
- Edges are chosen for only top ten nearest neighbors in feature space to ensure run time of $O(n \log n)$ where n is number of pixels.





Results





What we have learned today?

- Introduction to segmentation and clustering
- Gestalt theory for perceptual grouping
- Agglomerative clustering
- Oversegmentation