Photo by Unsplash user @foodiesfeed

BBM406
Fundamentals of Machine Learning

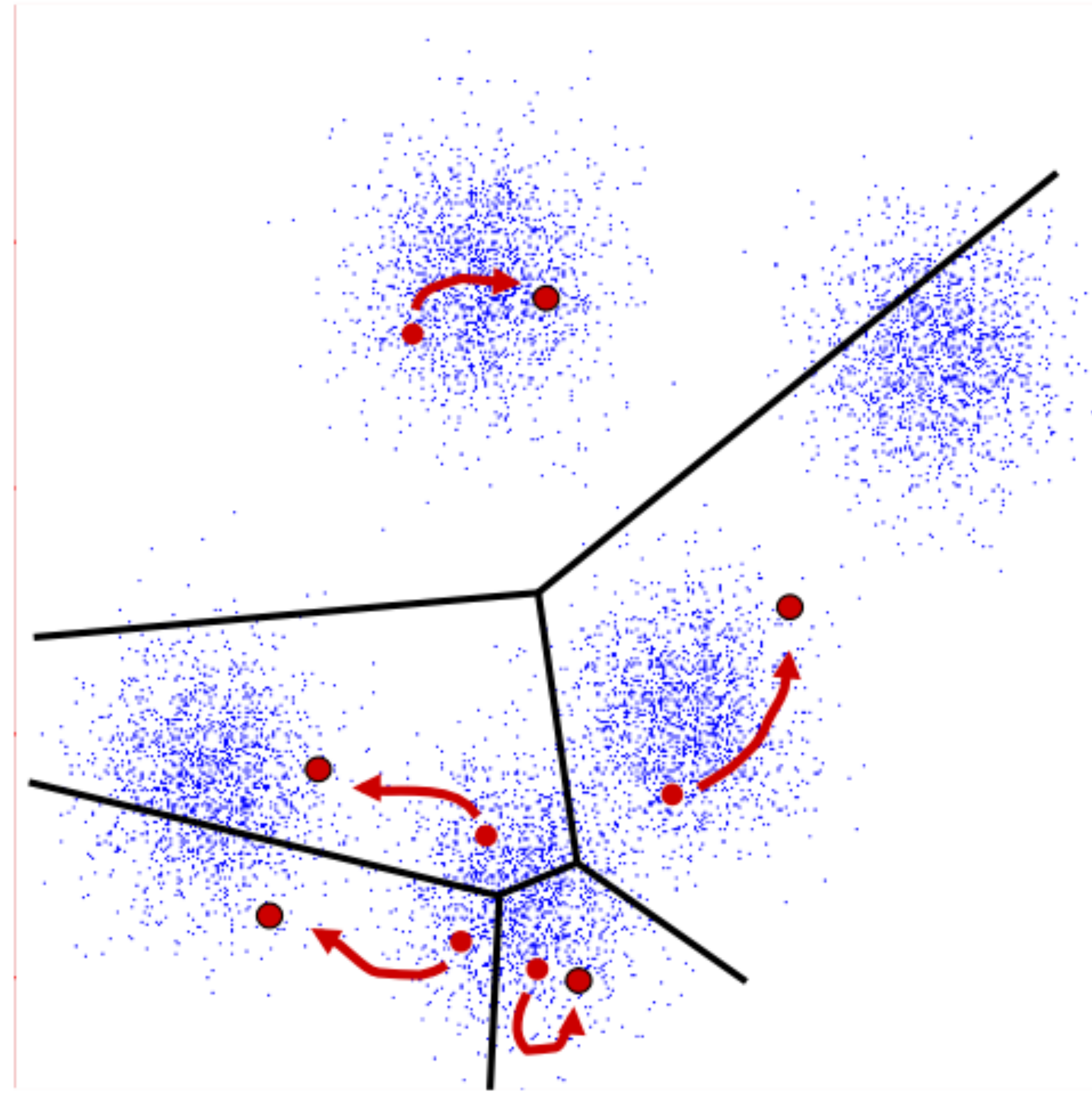
Lecture 22:
K-Means Example Applications
Spectral clustering
Hierarchical clustering
What is a good clustering?

HACETTEPE UNIVERSITY COMPUTER VISION LAB

Erkut Erdem // Hacettepe University // Spring 2021

Last time… **K-Means**

- An iterative clustering algorithm
  - Initialize: Pick K random points as cluster centers (means)
  - Alternate:
    - Assign data instances to closest mean
    - Assign each mean to the average of its assigned points
  - Stop when no points' assignments change



slide by David Sontag

Today

- K-Means Example Applications
- Spectral clustering
- Hierarchical clustering

K-Means
Example Applications

Example: K-Means for Segmentation

K=2

Goal of Segmentation is to partition an image into regions each of which has reasonably homogenous visual appearance.

Original

slide by David Sontag

Example: K-Means for Segmentation

K=2

K=3

Original

slide by David Sontag

Example: K-Means for Segmentation

K=2 K=3 K=10 Original







slide by David Sontag

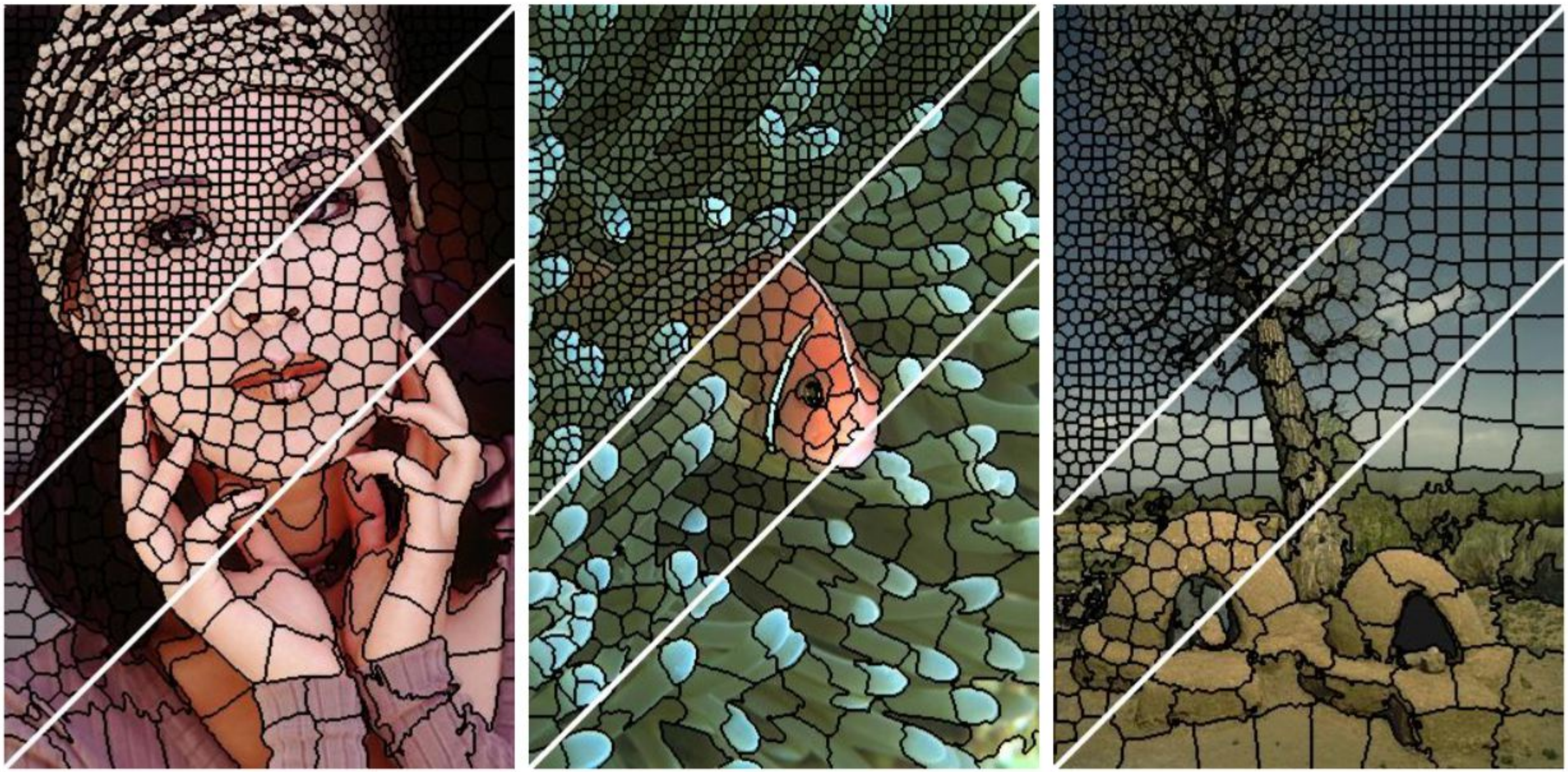
Example: Vector quantization

FIGURE 14.9. *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a* 1024×1024 *grayscale image at 8 bits per pixel. The center image is the result of* 2×2 *block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*

[Figure from Hastie *et al.* book]

slide by David Sontag

Example: Simple Linear Iterative Clustering (SLIC) superpixels



$$\Psi(x, y) = \begin{bmatrix} \lambda x \\ \lambda y \\ I(x, y) \end{bmatrix}$$

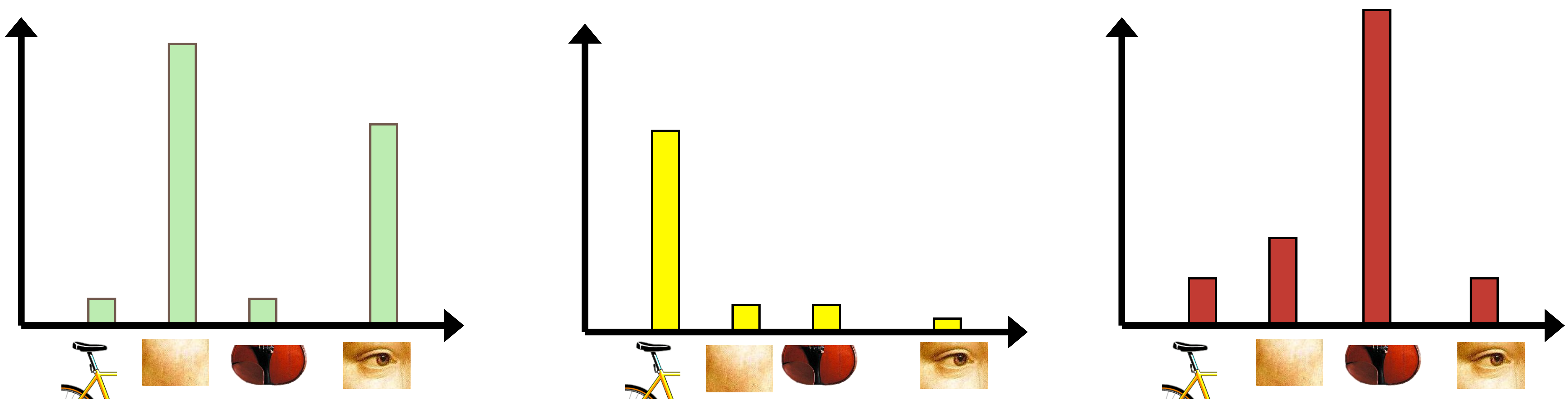
λ: spatial regularization parameter

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE T-PAMI, 2012

Bag of Words model

aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

slide by Carlos Guestrin

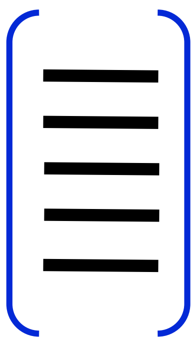


slide by Fei Fei Li

Object → **Bag of ‘words’**

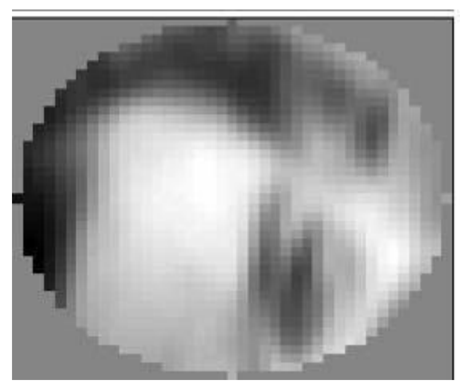
slide by Fei Fei Li

Interest Point Features

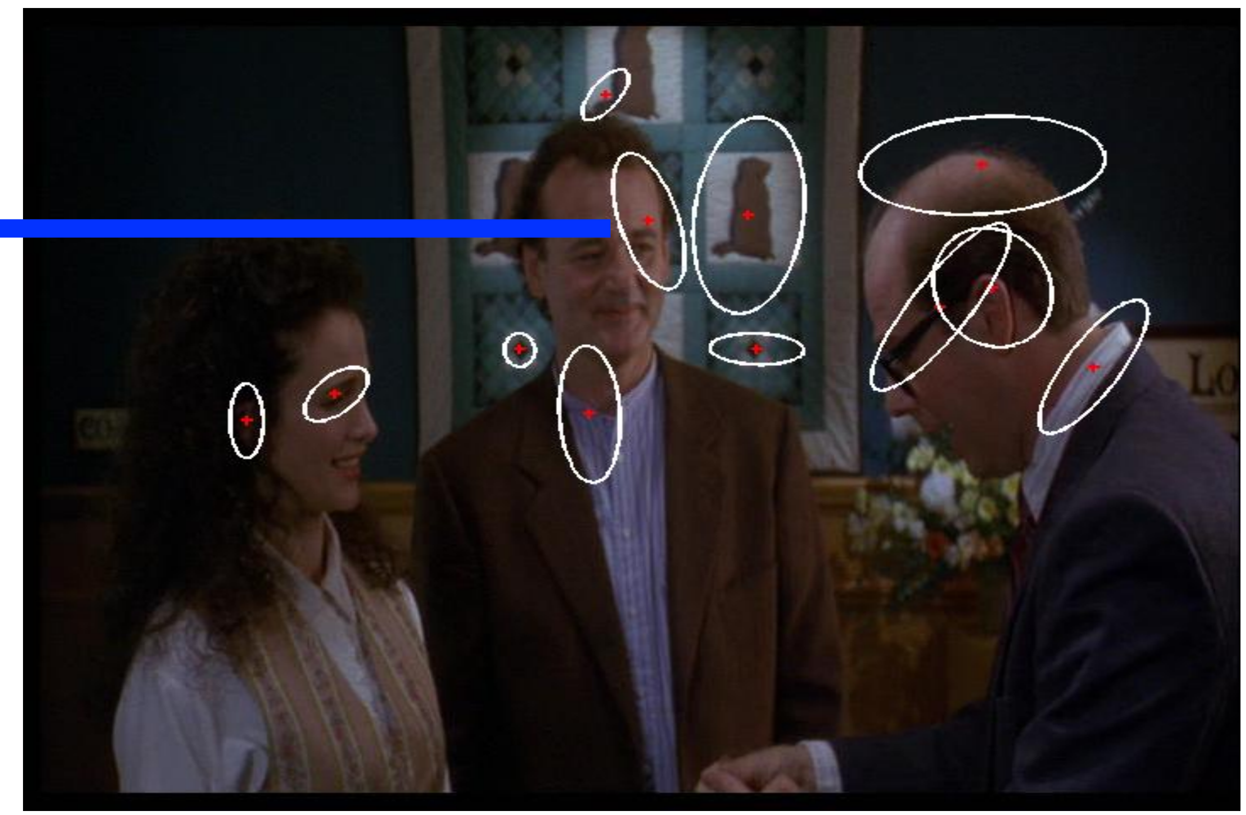


Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

[Mikojaczyk and Schmid '02]

[Matas et al. '02]

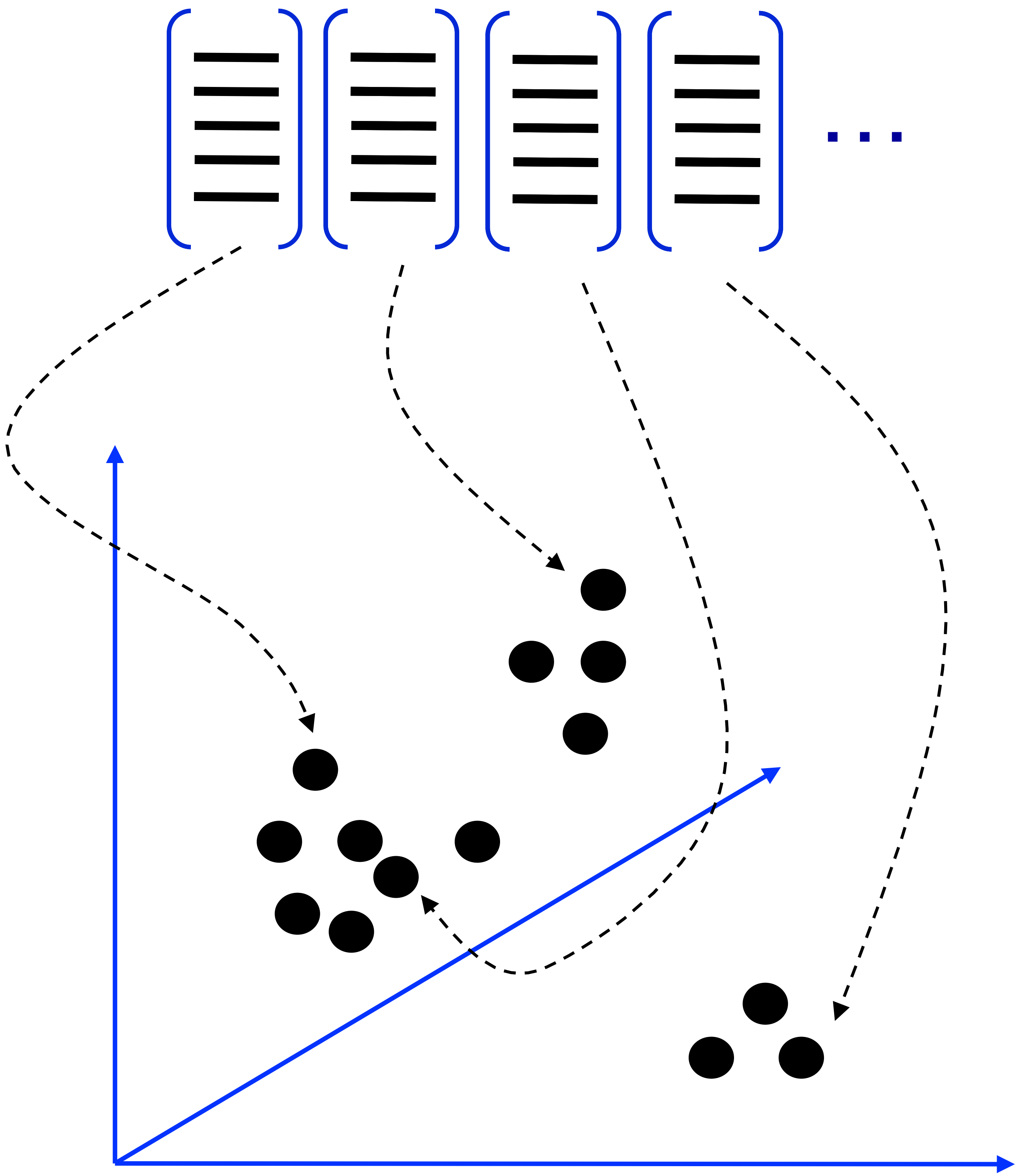
[Sivic et al. '03]

slide by Josef Sivic

Patch Features

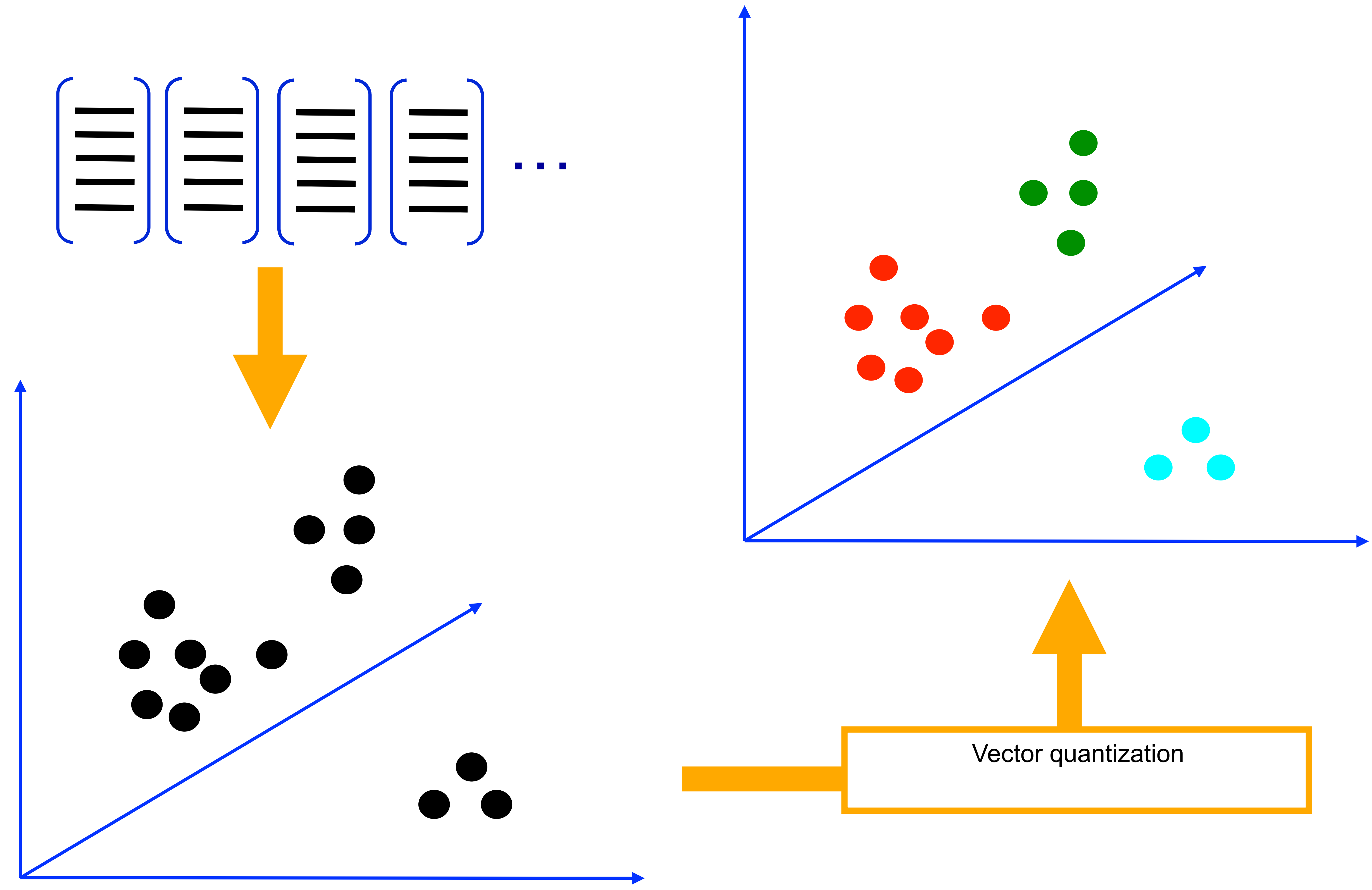
slide by Josef Sivic

Dictionary Formation



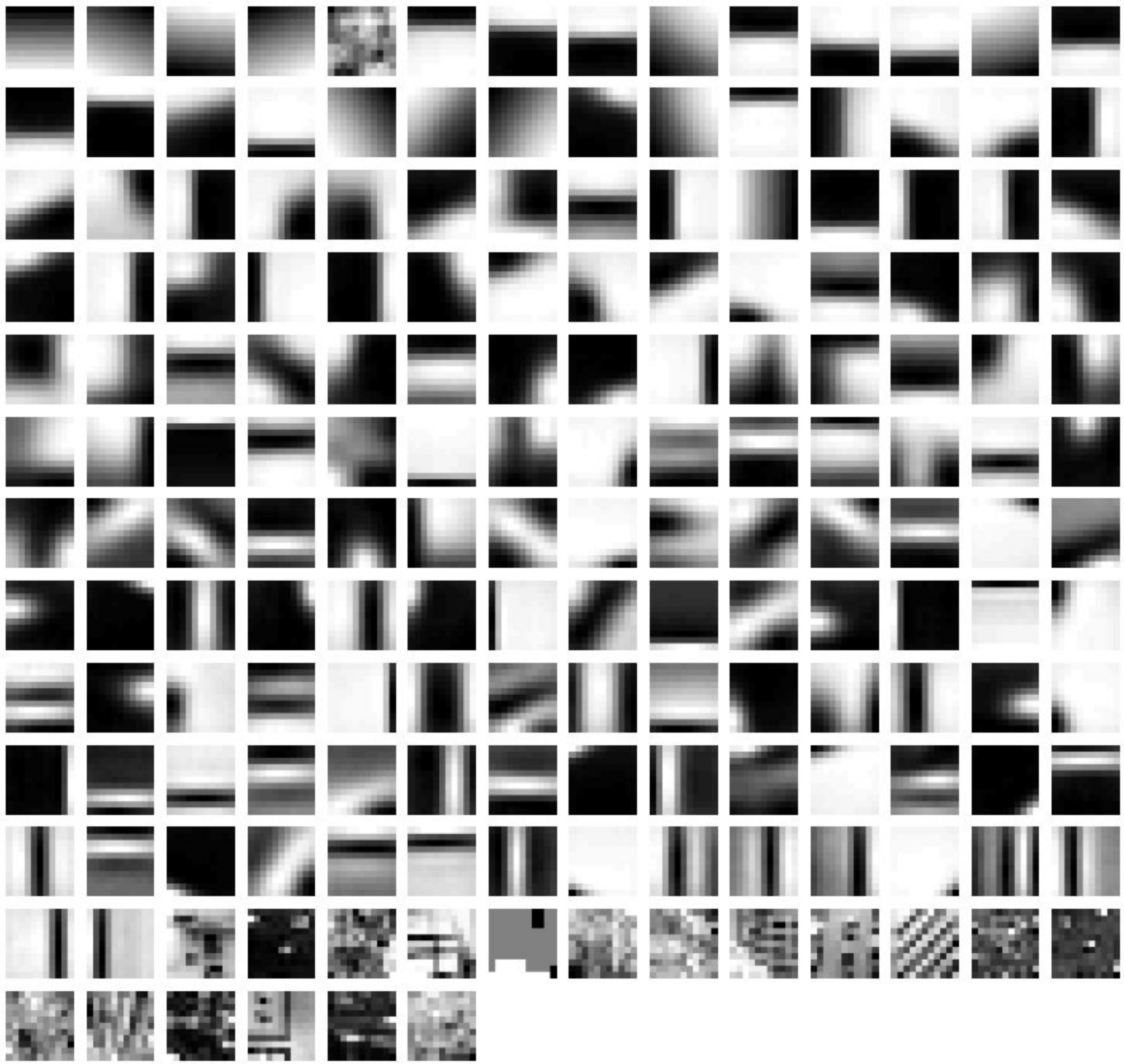
slide by Josef Sivic

Clustering (usually K-means)



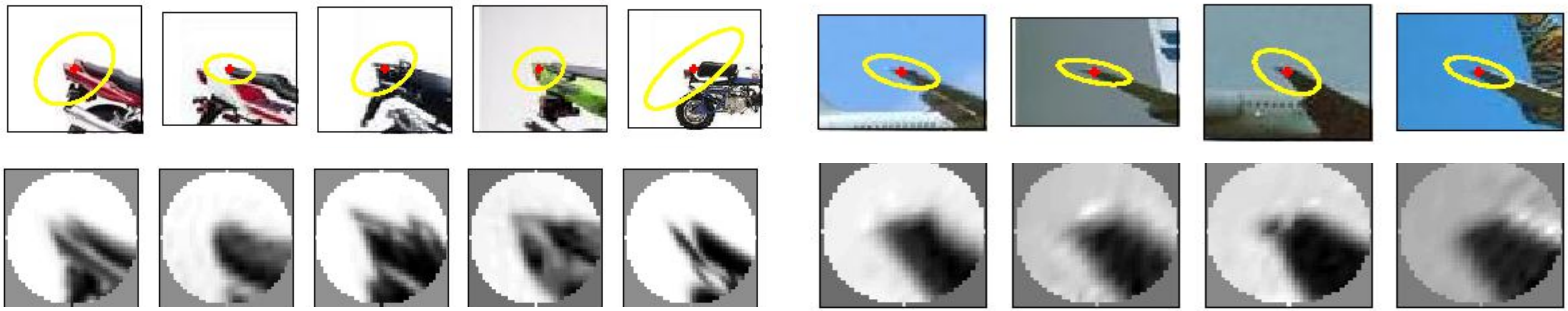
slide by Josef Sivic

Clustered Image Patches

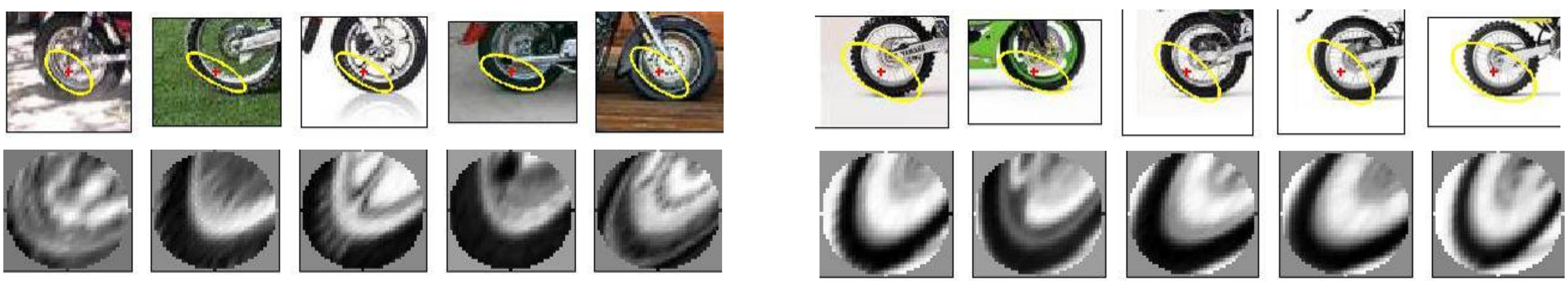


slide by Fei Fei Li

Visual synonyms and polysemy



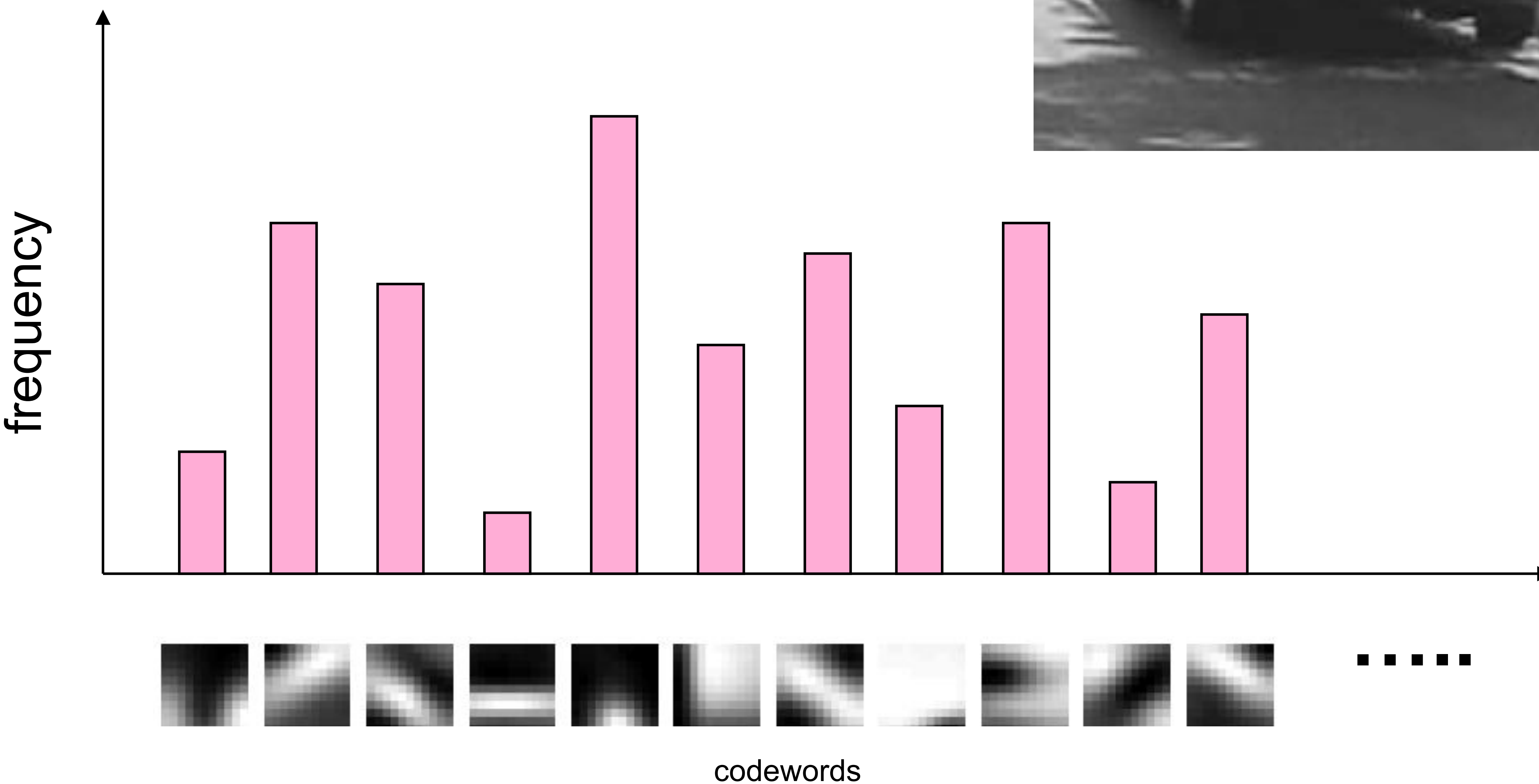
Visual Polysemy. Single visual word occurring on different (but locally similar) parts on different object categories.



Visual Synonyms. Two different visual words representing a similar part of an object (wheel of a motorbike).

slide by Andrew Zisserman

Image Representation



slide by Fei Fei Li

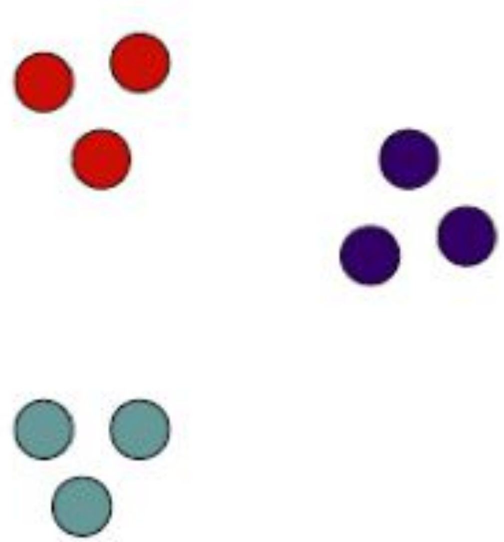
Spectral clustering

Graph-Theoretic Clustering

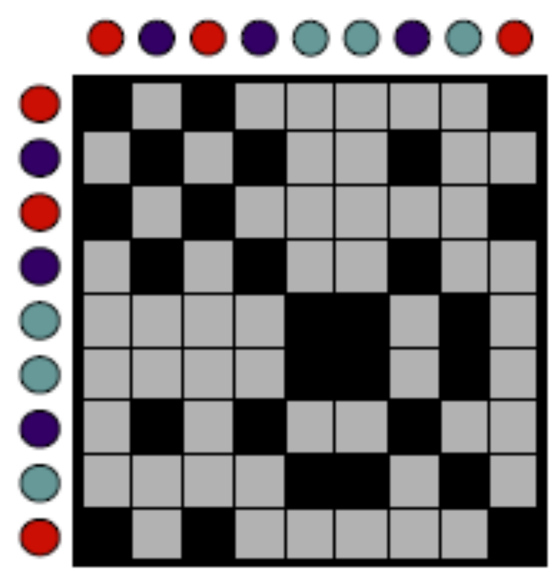
Goal: Given data points $x_1, \ldots, x_n$ and similarities $W(x_i, x_j)$, partition the data into groups so that points in a group are similar and points in different groups are dissimilar.

Similarity Graph: G(V,E,W)

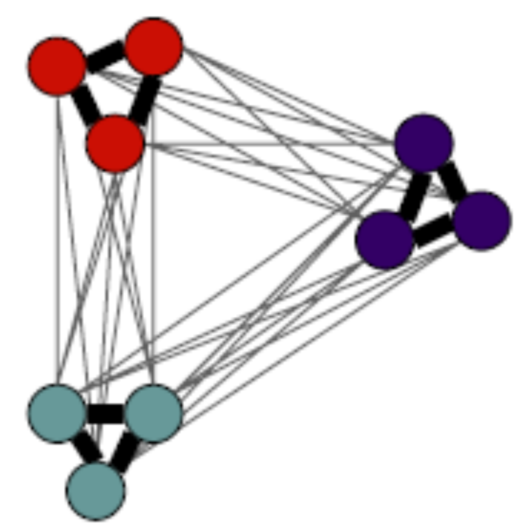
V – Vertices (Data points)
E – Edge if similarity > 0
W - Edge weights (similarities)



Data



Similarities

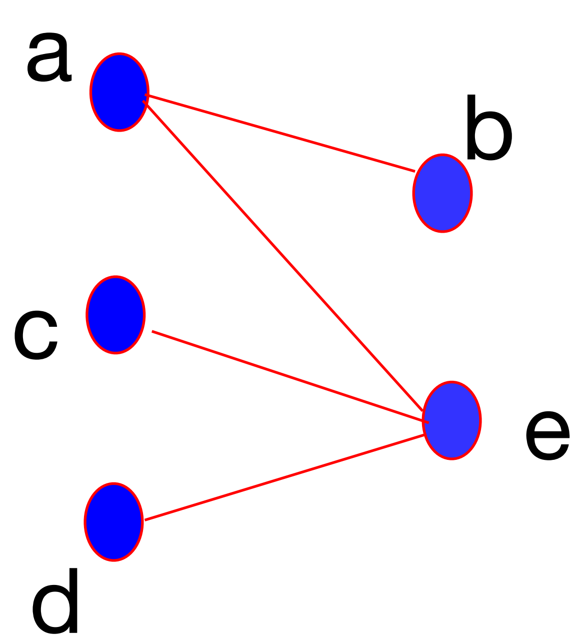


Similarity graph

Partition the graph so that edges within a group have large weights and edges across groups have small weights.

slide by Aarti Singh

Graphs Representations

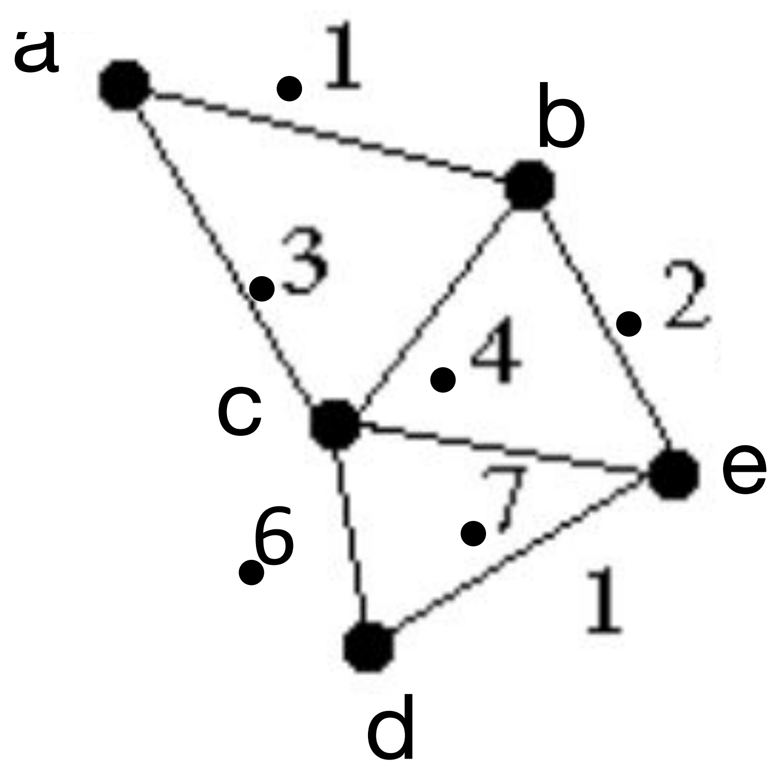


$$
\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}
\begin{array}{c}
\begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}
$$

Adjacency Matrix

slide by Bill Freeman and Antonio Torralba

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 cluster

slide by Bill Freeman and Antonio Torralba

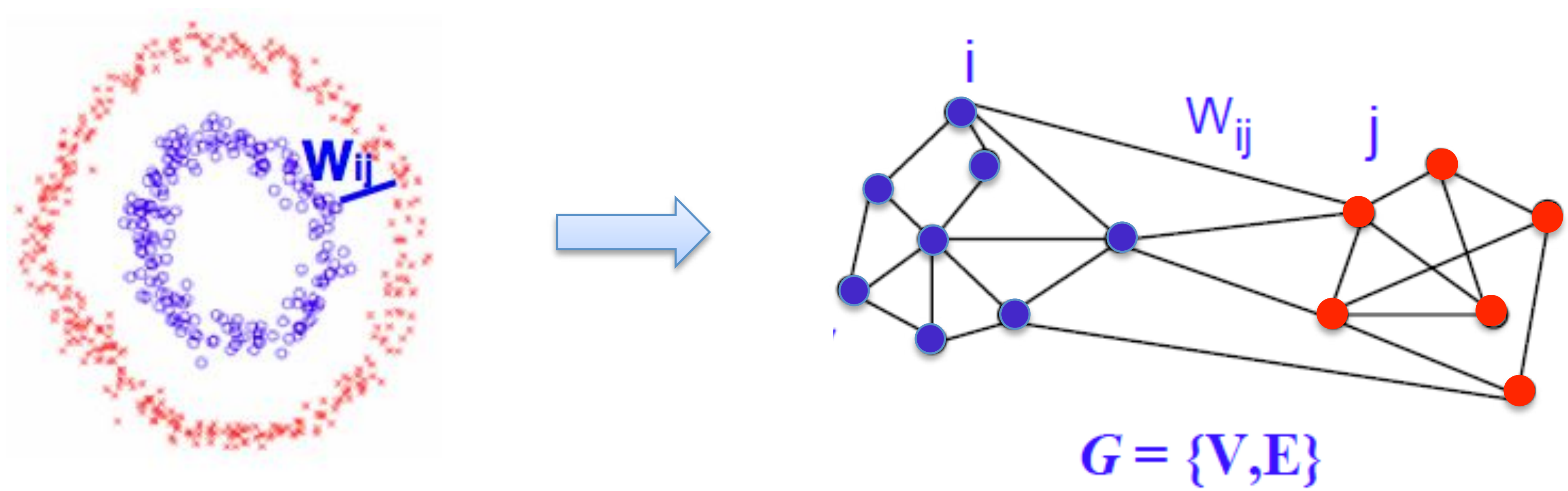
Similarity graph construction

- Similarity Graphs: Model local neighborhood relations between data points
- E.g. epsilon-NN

$$W_{ij} = \begin{cases} 1 & \|x_i - x_j\| \leq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

Controls size of neighborhood

or mutual k-NN graph (W_{ij} = 1 if x_i or x_j is k nearest neighbor of the other)



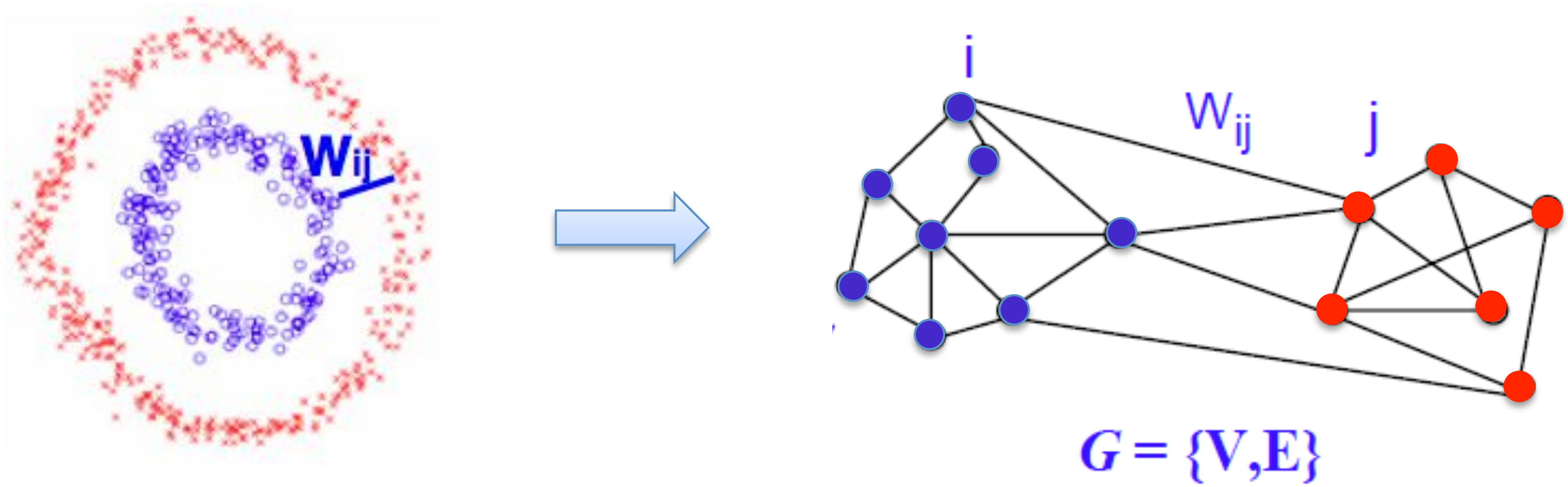
slide by Aarti Singh

Similarity graph construction

- Similarity Graphs: Model local neighborhood relations between data points
- E.g. Gaussian kernel similarity function

$$W_{ij} = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

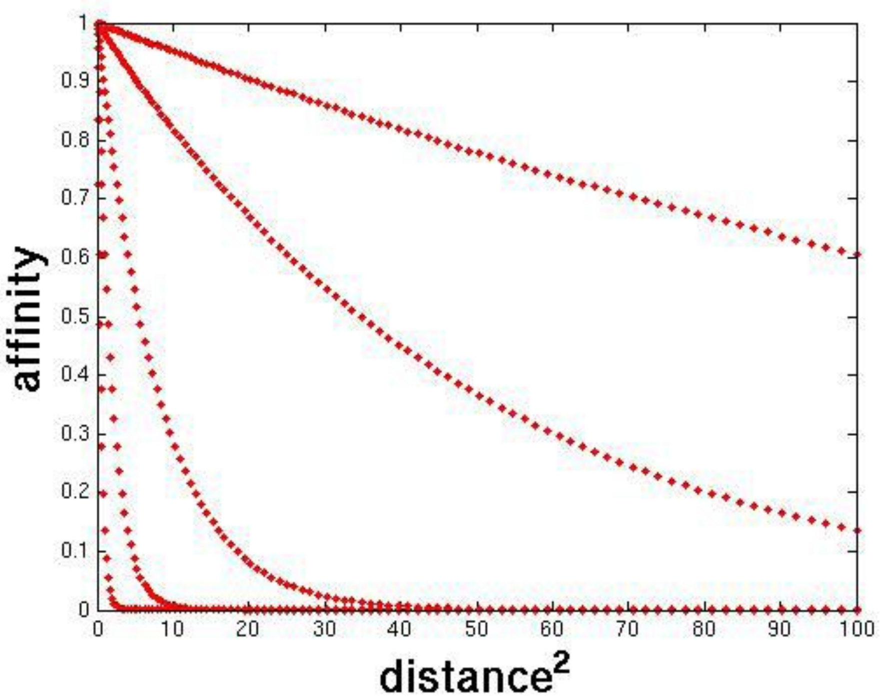
Controls size of neighborhood

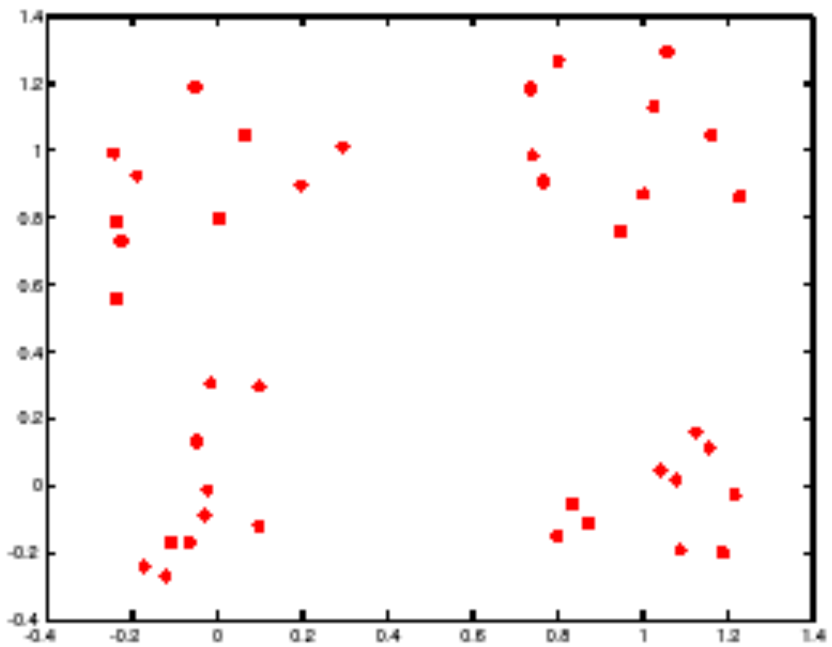


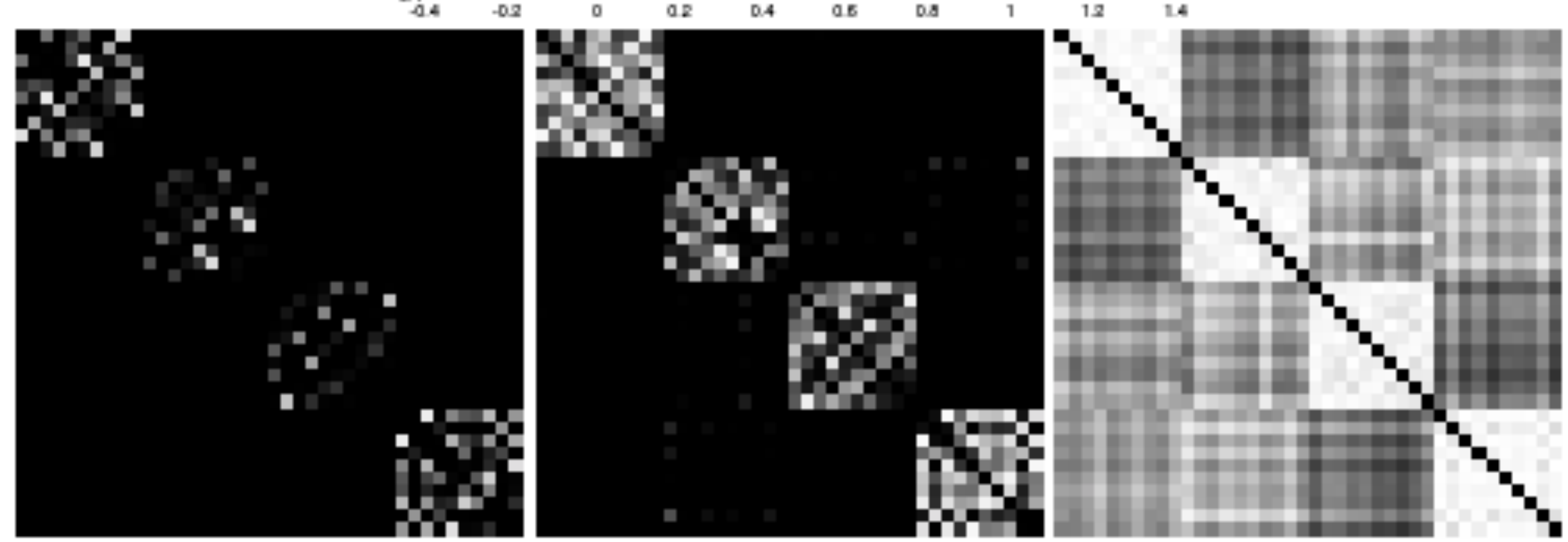
slide by Aarti Singh

Scale affects affinity

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







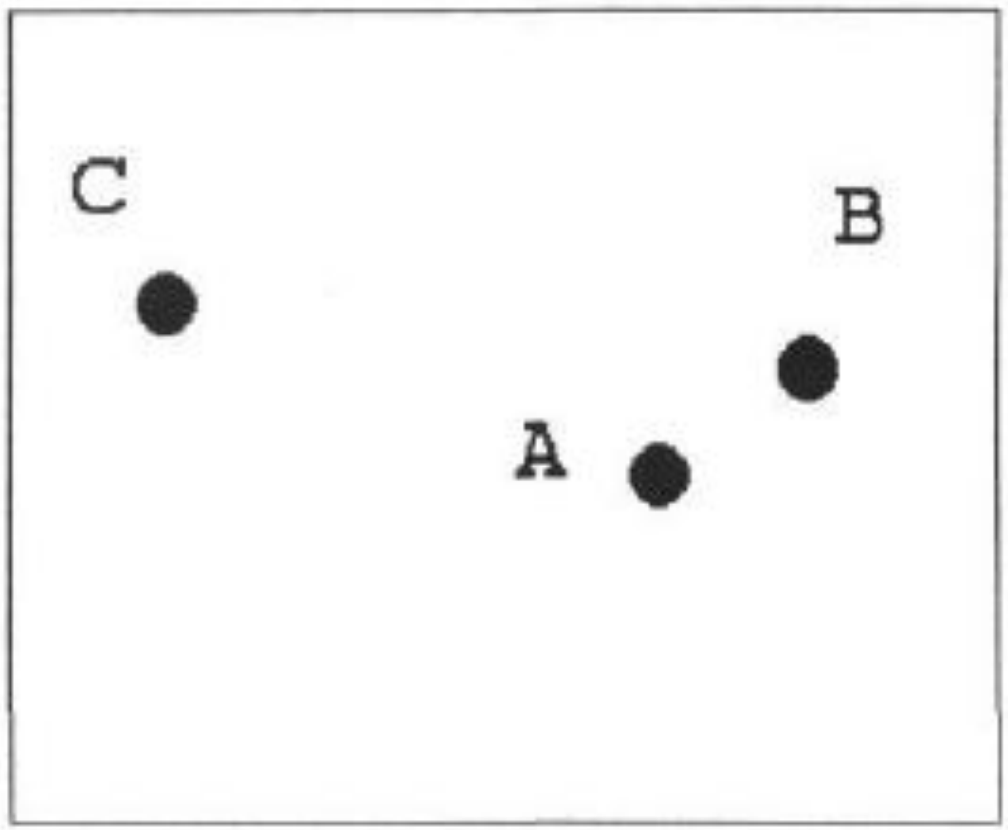
slide by Svetlana Lazebnik

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=

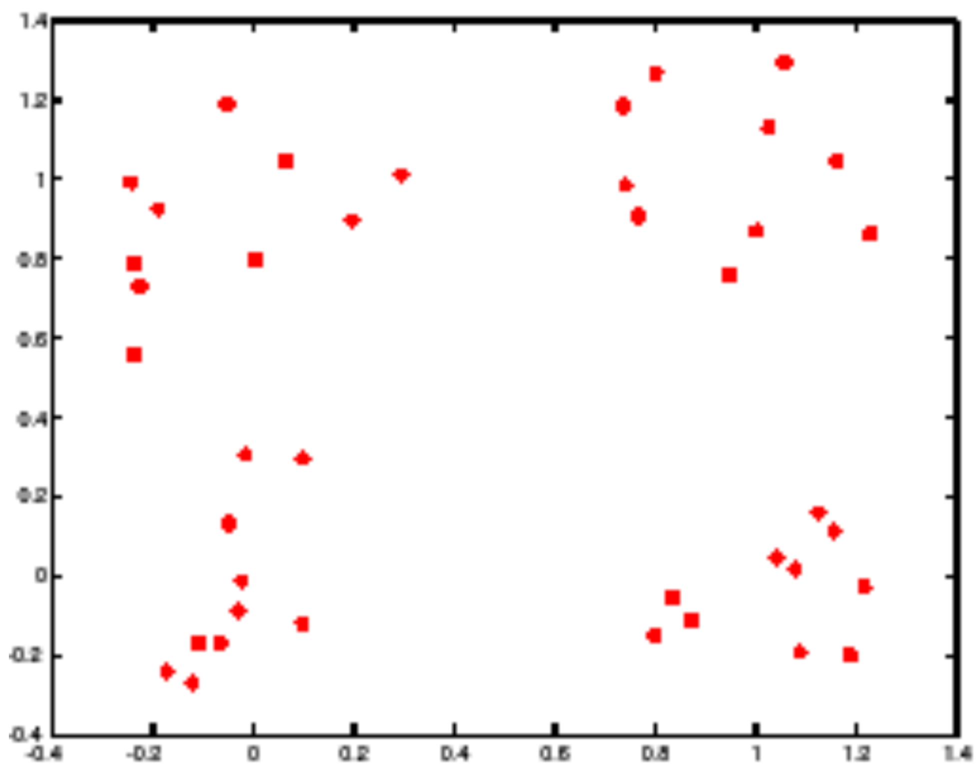
	A	B	C
A	1.00	0.63	0.03
B	0.63	1.00	0.0
C	0.03	0.0	1.00

The eigenvectors of W are:

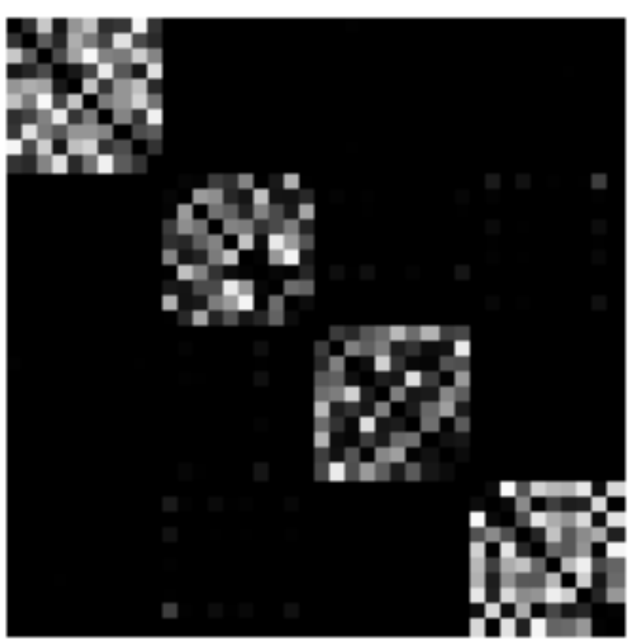
	E_1	E_2	E_3
Eigenvalues	1.63	1.00	0.37
A	-0.71	-0.01	0.71
B	-0.71	-0.05	-0.71
C	-0.04	1.00	-0.03

The first 2 eigenvectors group the points as desired…

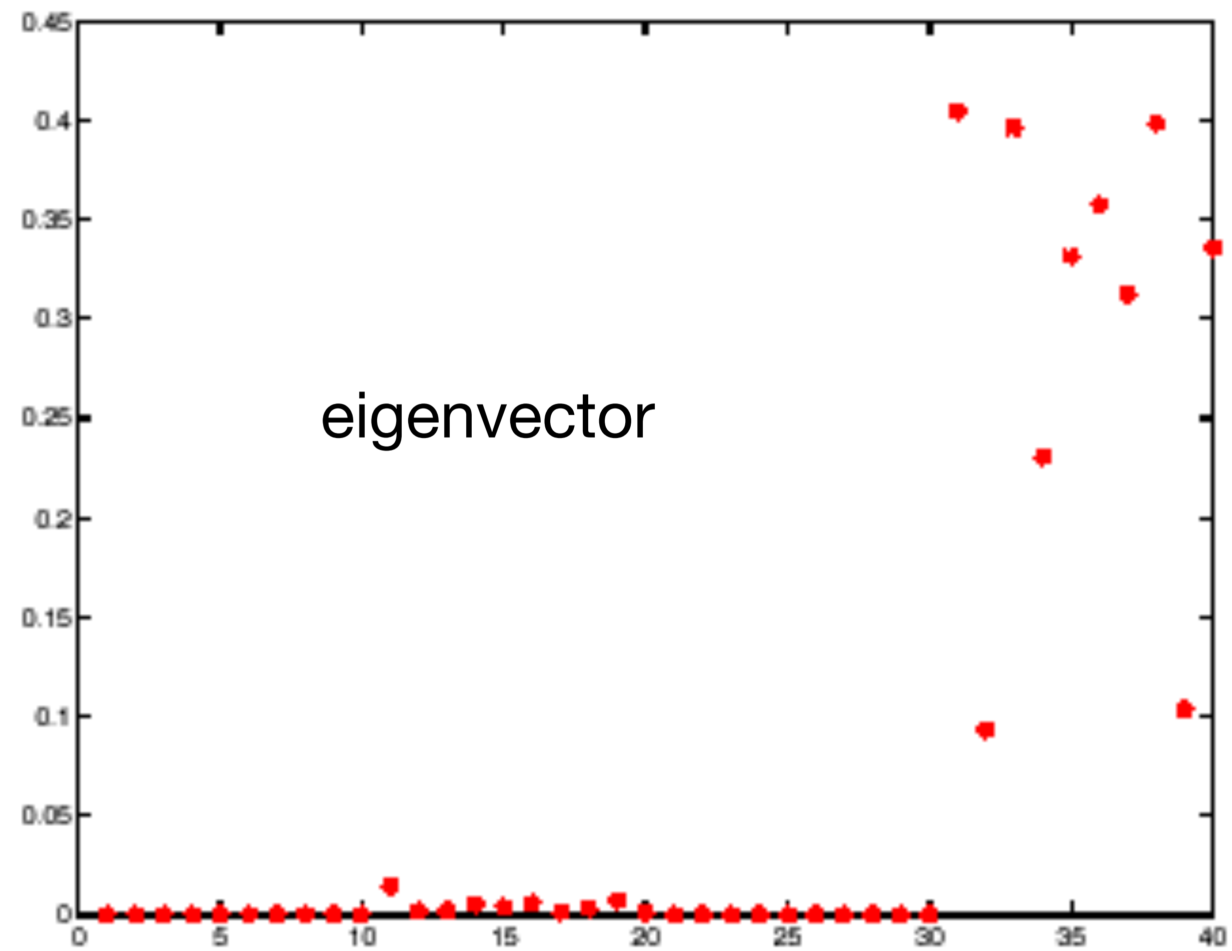
Example eigenvector



points

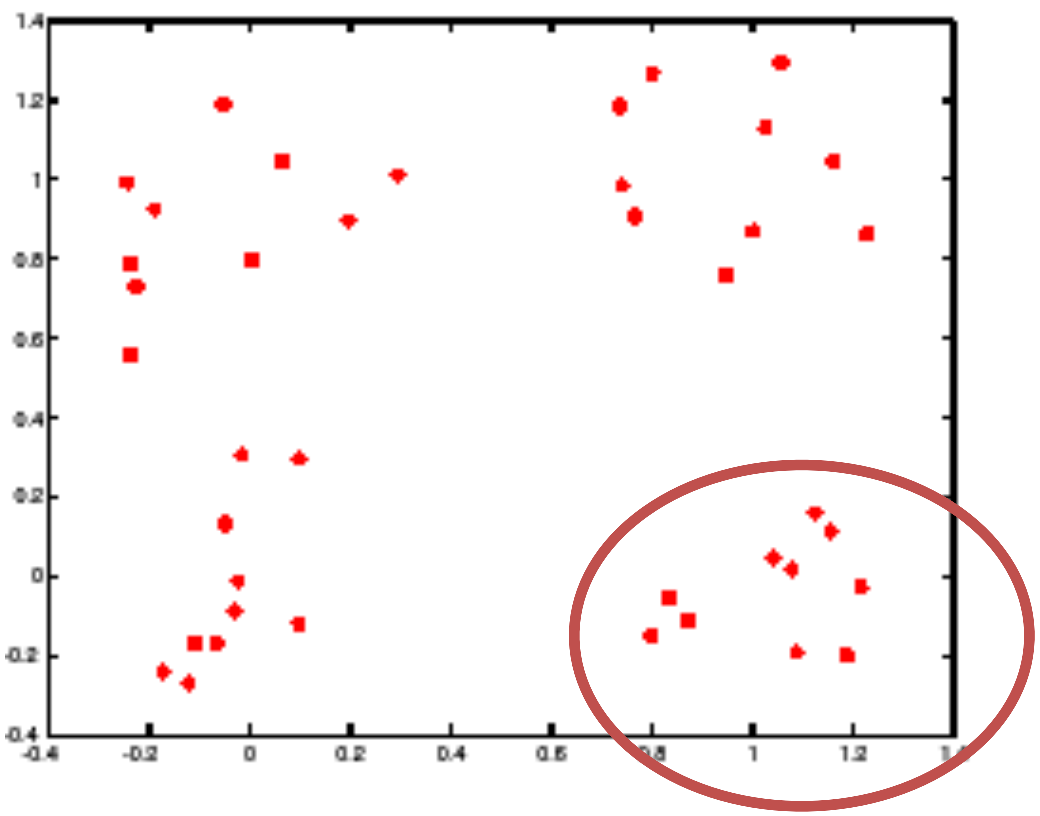


Affinity matrix

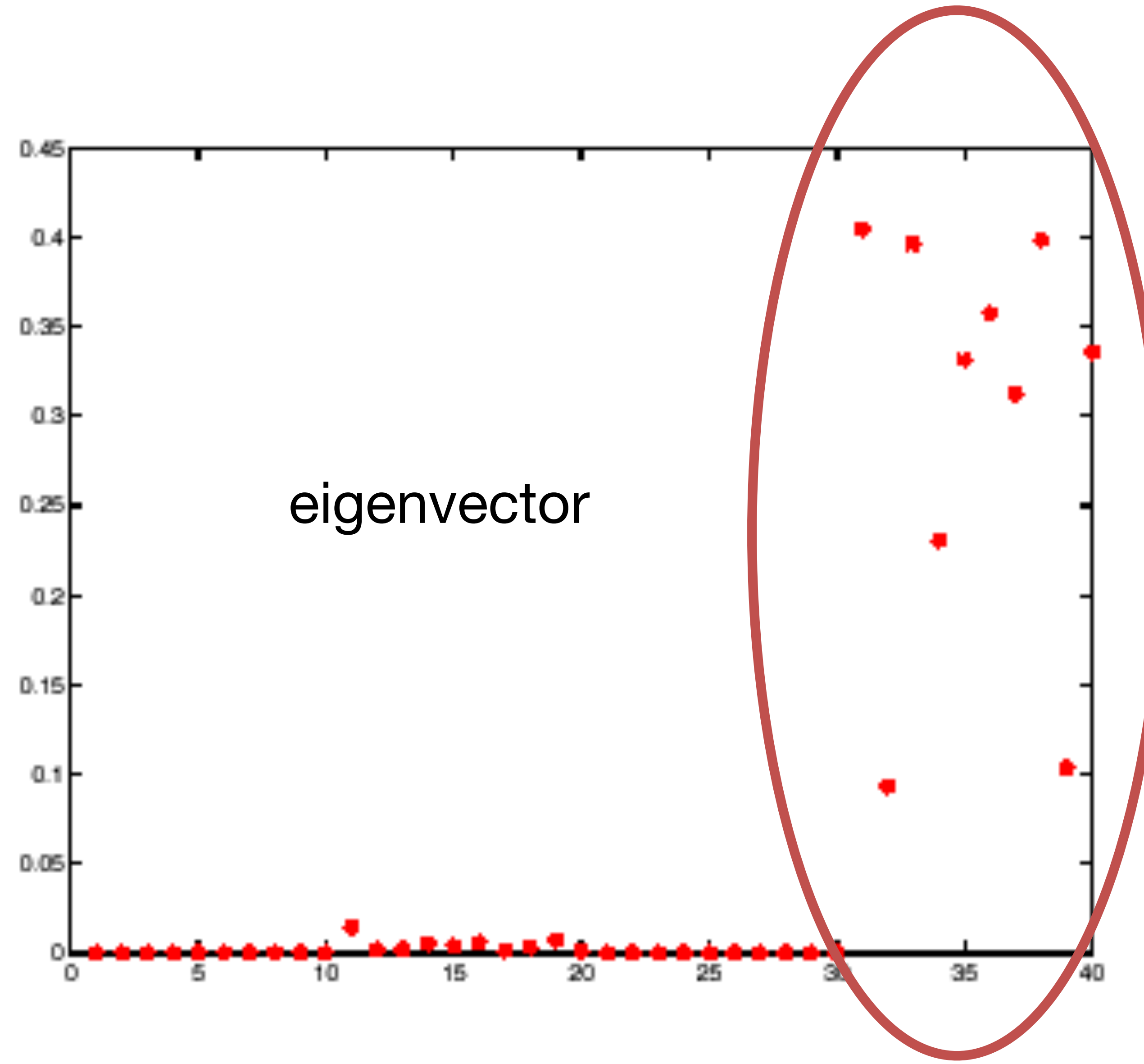


slide by Bill Freeman and Antonio Torralba

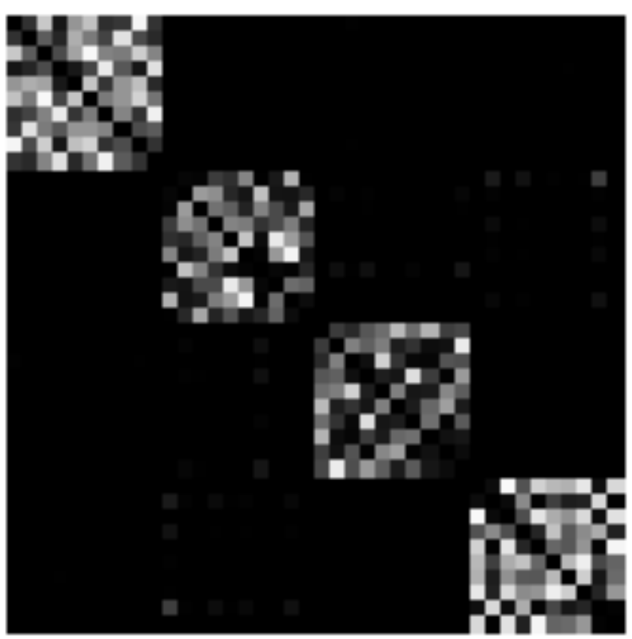
Example eigenvector



points



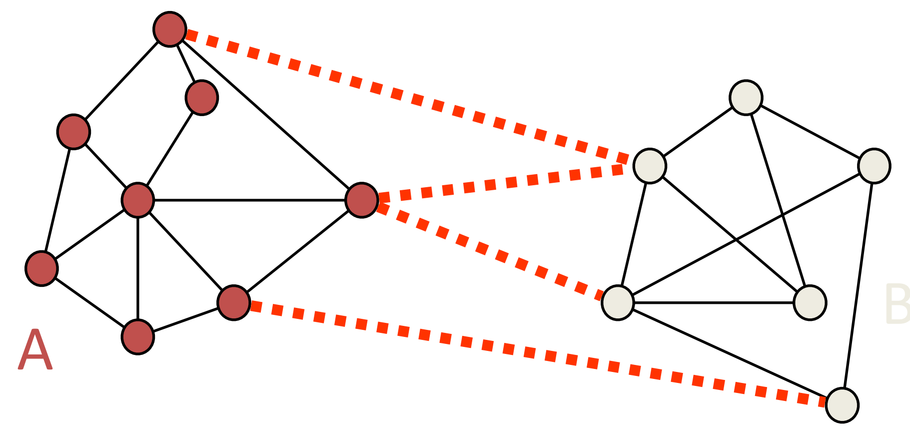
eigenvector



Affinity matrix

slide by Bill Freeman and Antonio Torralba

Graph cut

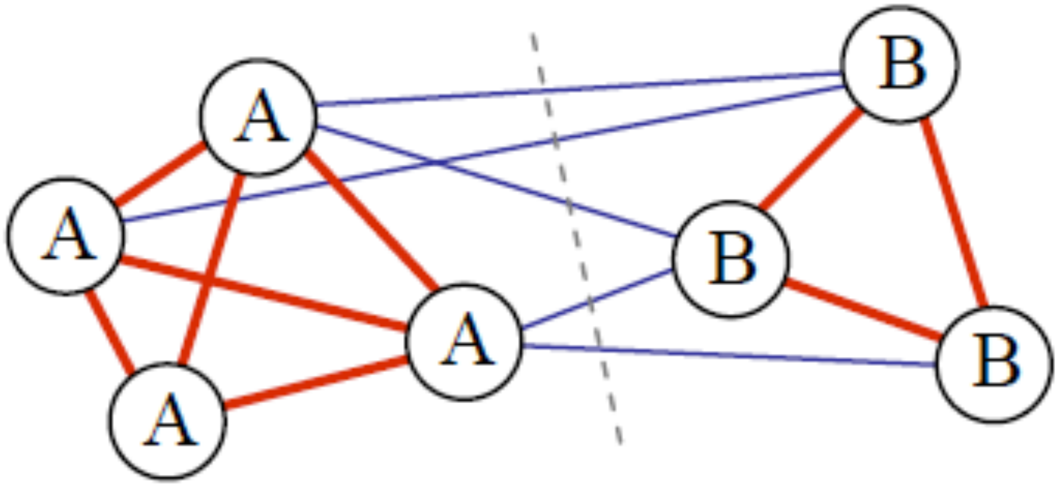


- Set of edges whose removal makes a graph disconnected
- Cost of a cut: sum of weights of cut edges
- A graph cut gives us a partition (clustering)
  - What is a "good" graph cut and how do we find one?

slide by Steven Seitz

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:

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

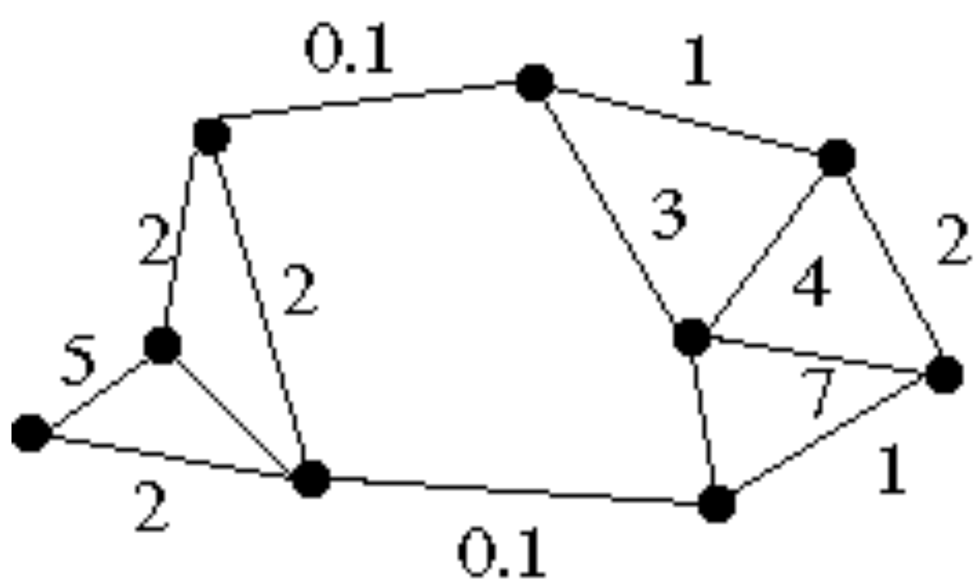
with $\mathrm{A} \cap \mathrm{B} = \emptyset$

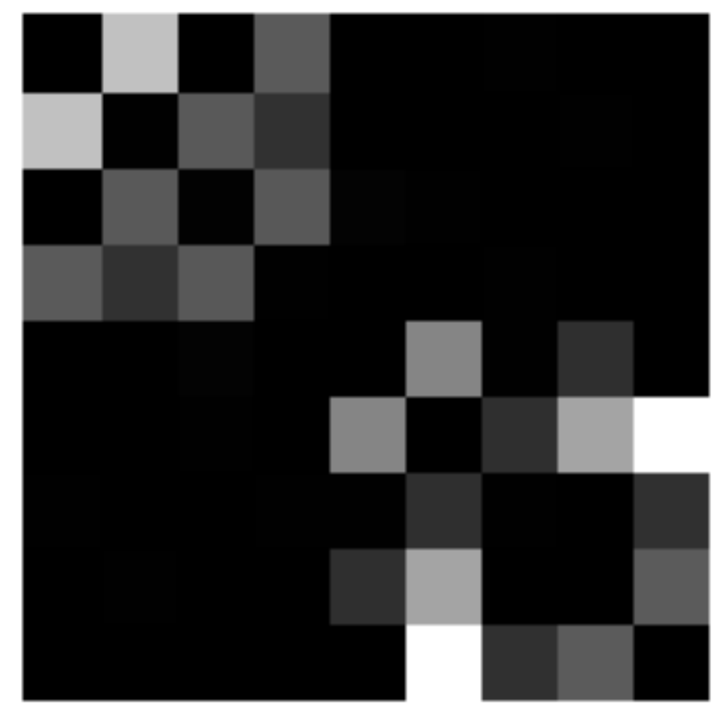
slide by Bill Freeman and Antonio Torralba

Minimum cut

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

Minimum cut example



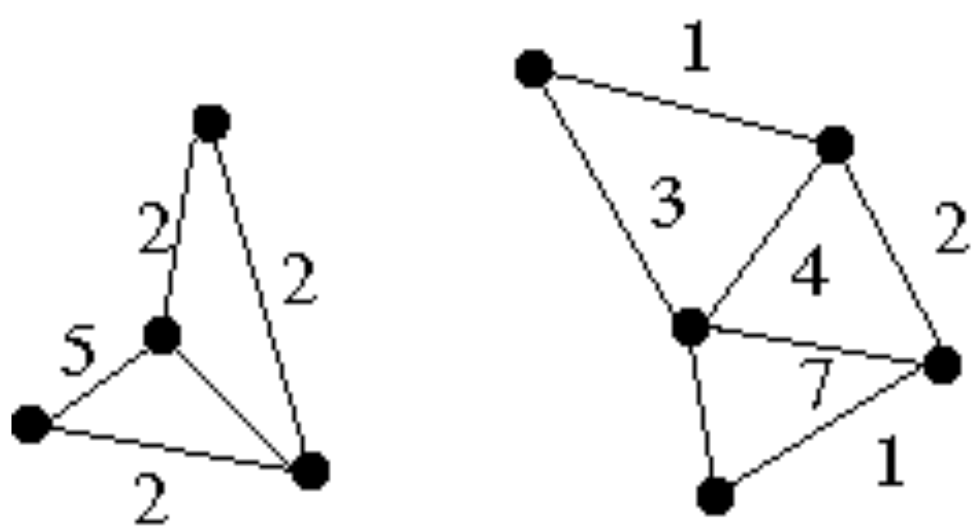


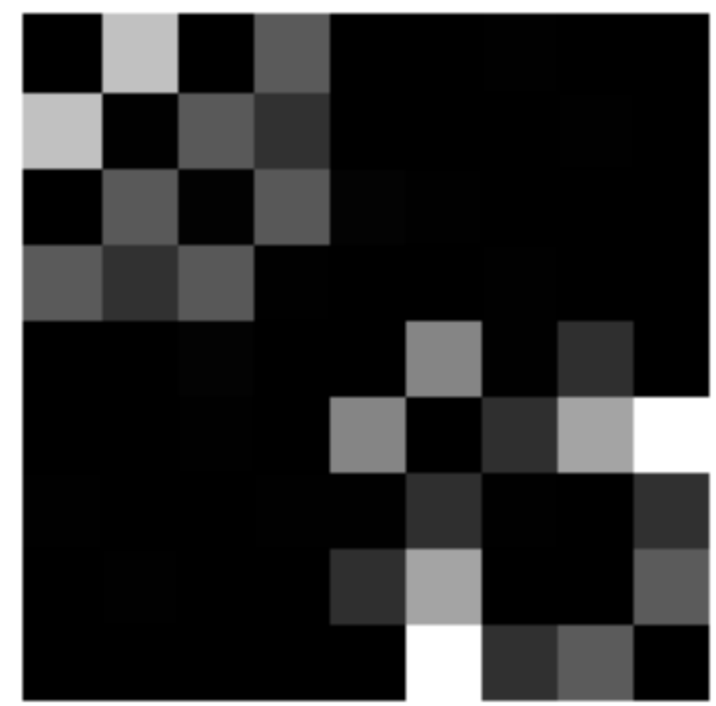
slide by Svetlana Lazebnik

Minimum cut

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

Minimum cut example

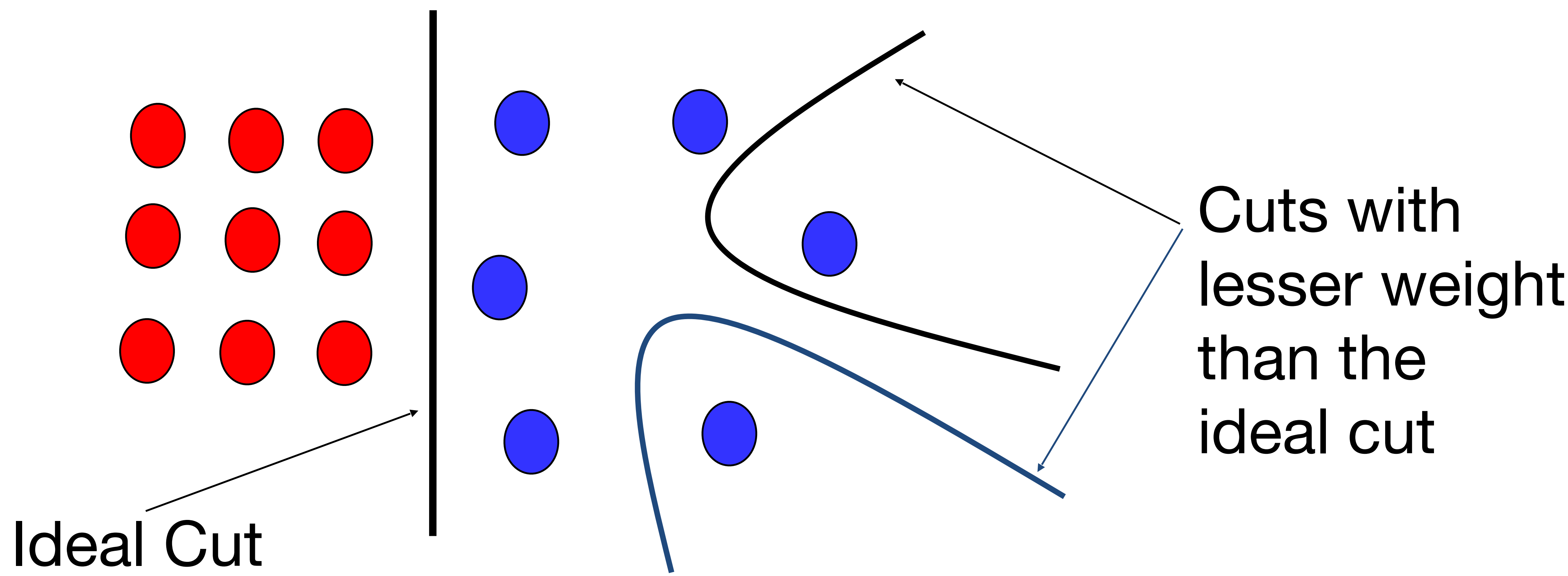




slide by Svetlana Lazebnik

Drawbacks of Minimum cut

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

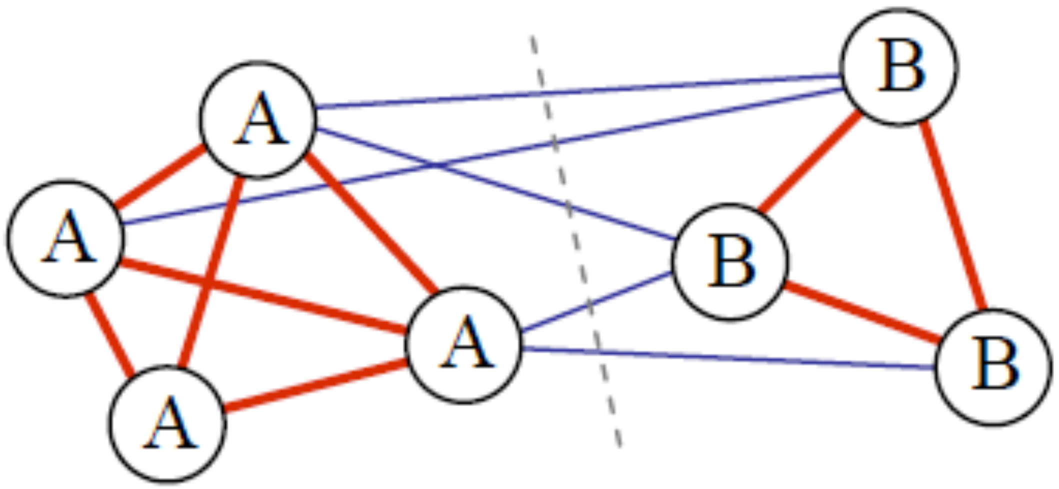


slide by Bill Freeman and Antonio Torralba

* Slide from Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Normalized cuts

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



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

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

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

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

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

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

A and B not necessarily disjoint

slide by Bill Freeman and Antonio Torralba

J. Shi and J. Malik. [Normalized cuts and image segmentation.]() PAMI 2000

Normalized cut

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

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

D-W: Graph Laplacian

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

slide by Svetlana Lazebnik

J. Shi and J. Malik. [Normalized cuts and image segmentation.]() PAMI 2000

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, aka the Fiedler vector
- Intuitively, the i-th entry of y can be viewed as a “soft” indication of the component membership of the i-th feature
  - Can use 0 or median value of the entries as the splitting point (threshold), or find threshold that minimizes the Ncut cost

slide by Svetlana Lazebnik

J. Shi and J. Malik. [Normalized cuts and image segmentation.](#) PAMI 2000

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.

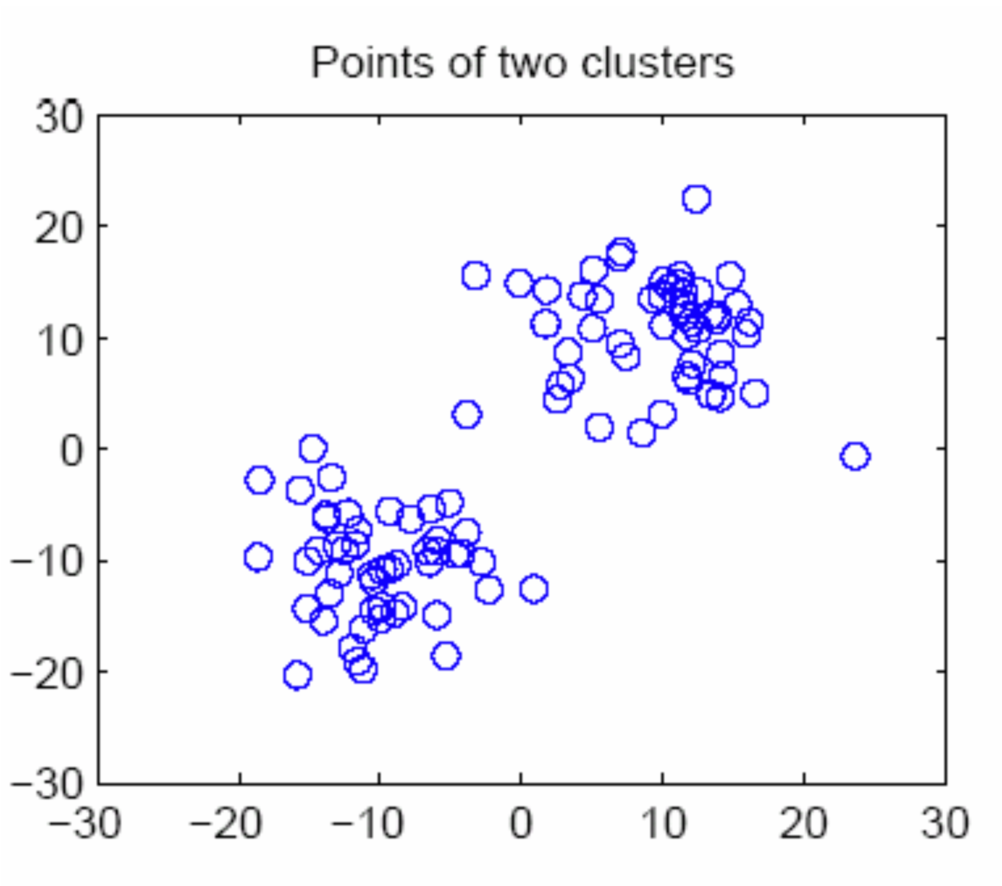
4. Decide if the current partition should be sub-divided, and recursively repartition the segmented parts if necessary.

slide by Bill Freeman and Antonio Torralba

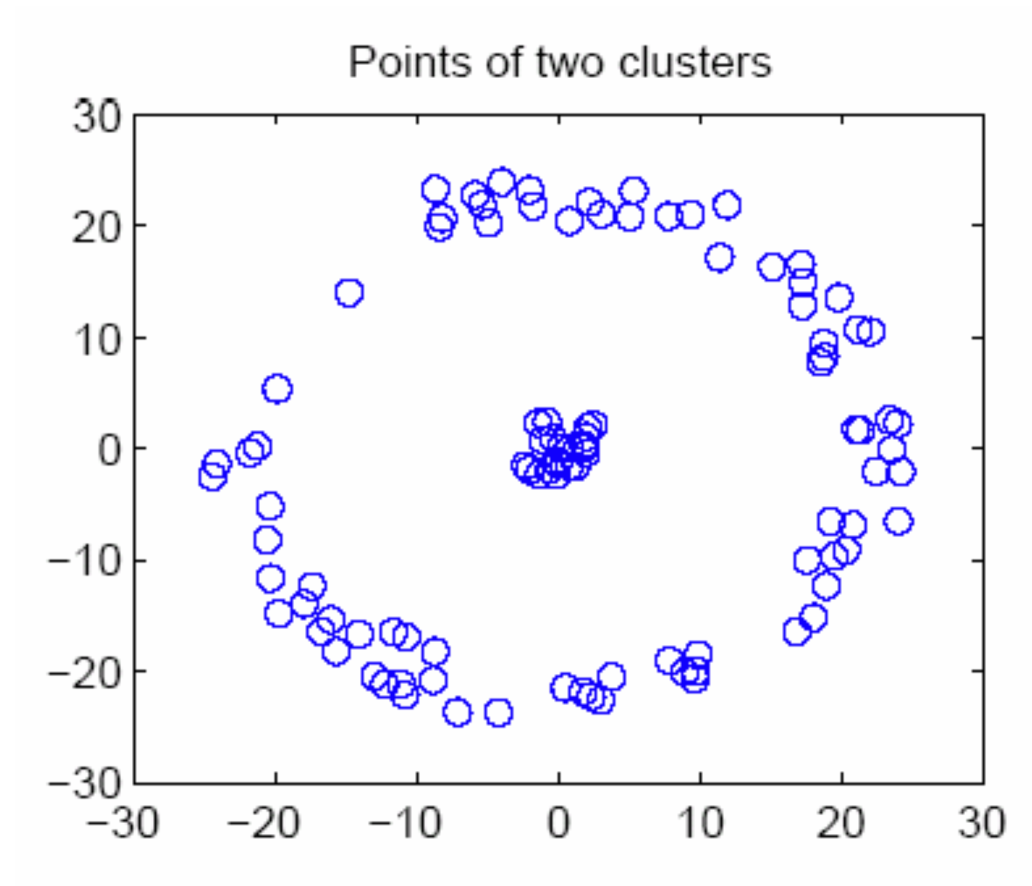
J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI 2000

K-Means vs. Spectral Clustering

- Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.



Both perform same

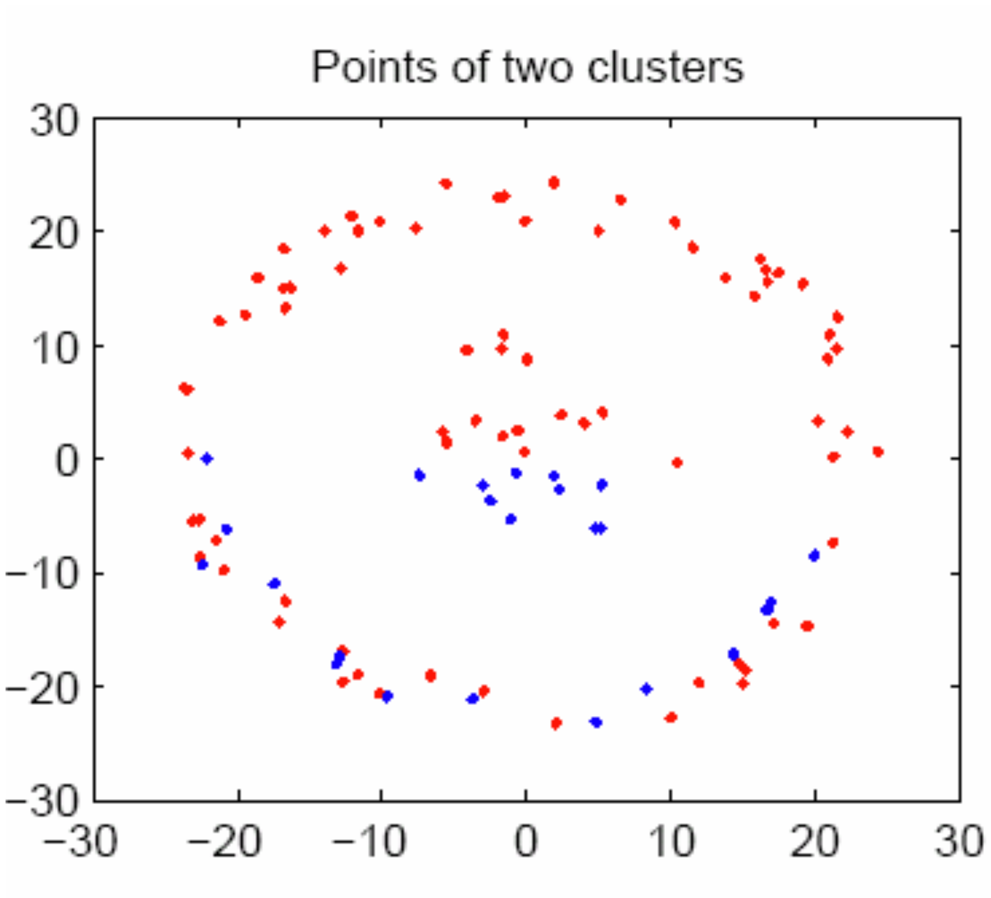


Spectral clustering is superior

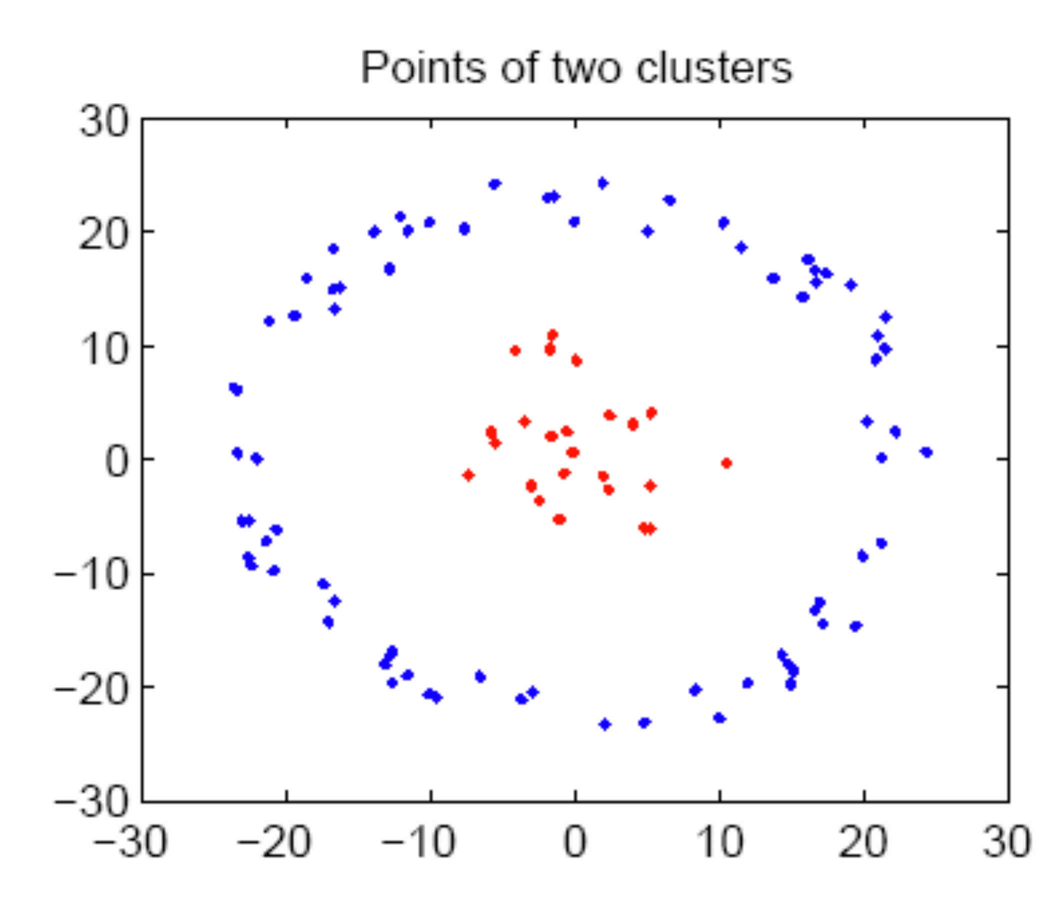
slide by Aarti Singh

K-Means vs. Spectral Clustering

- Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.



k-means output



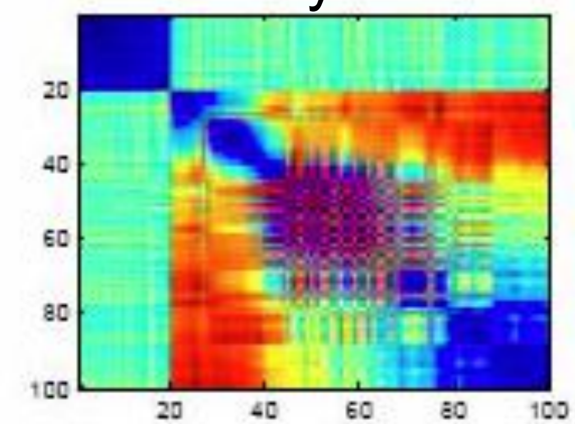
Spectral clustering output

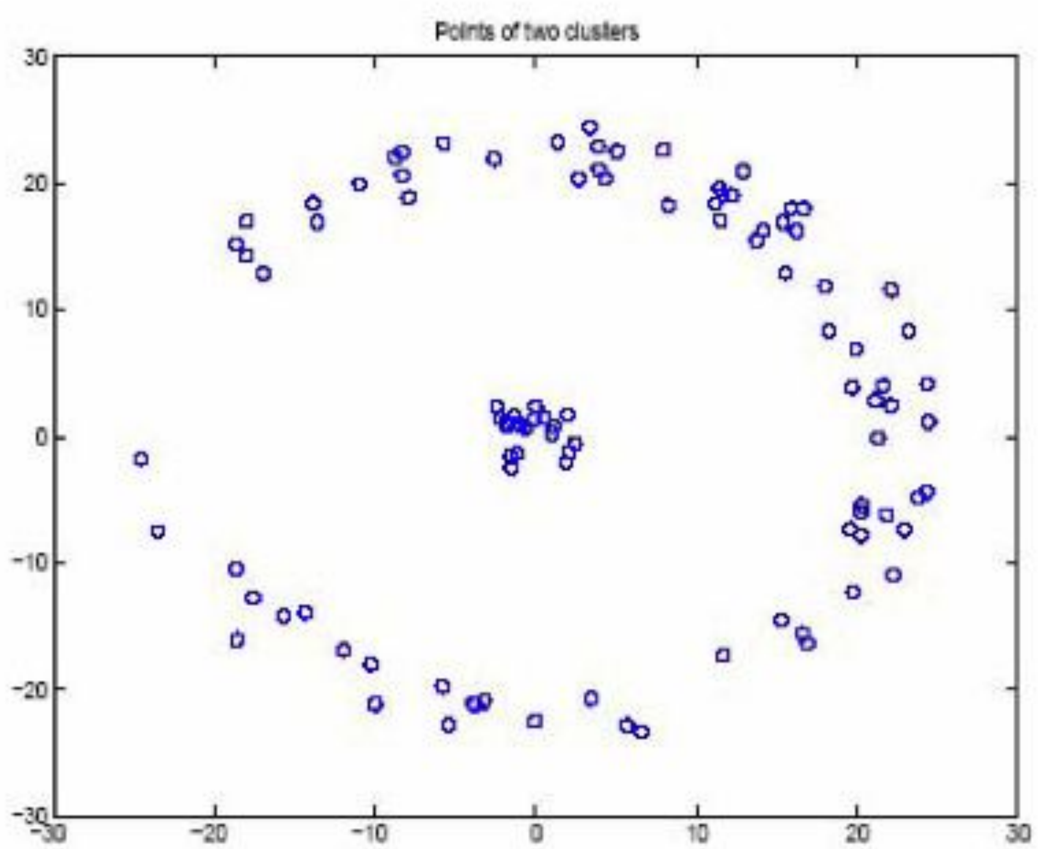
slide by Aarti Singh

K-Means vs. Spectral Clustering

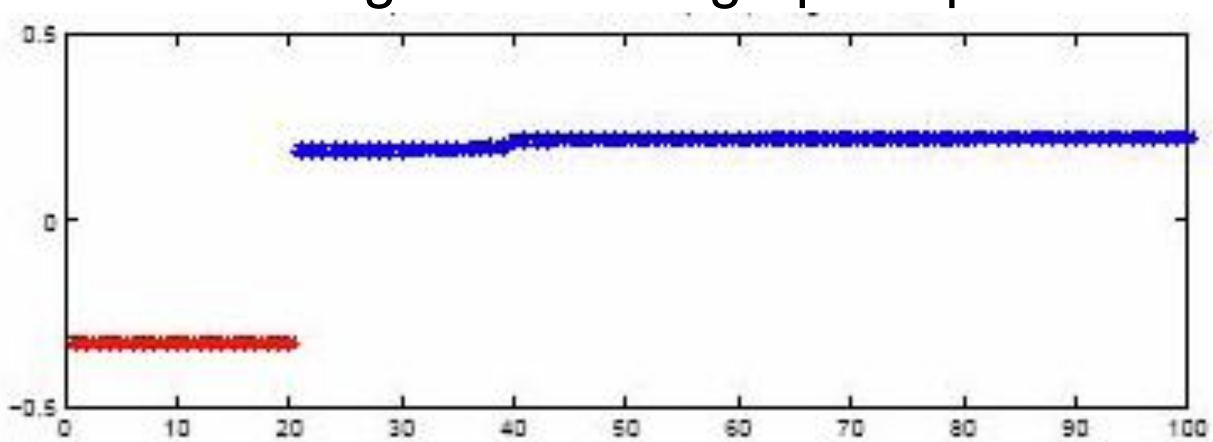
- Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.

Similarity matrix



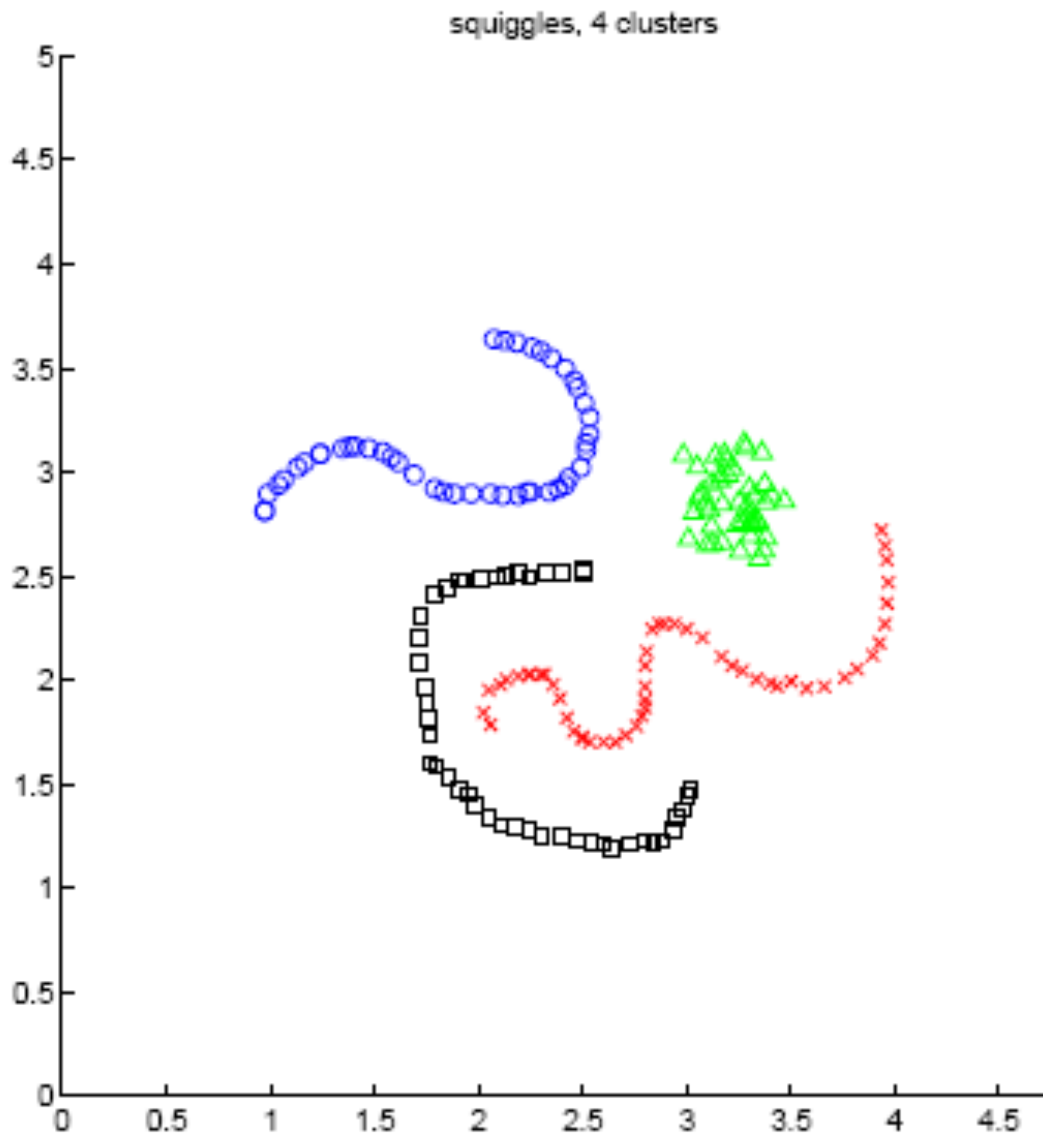


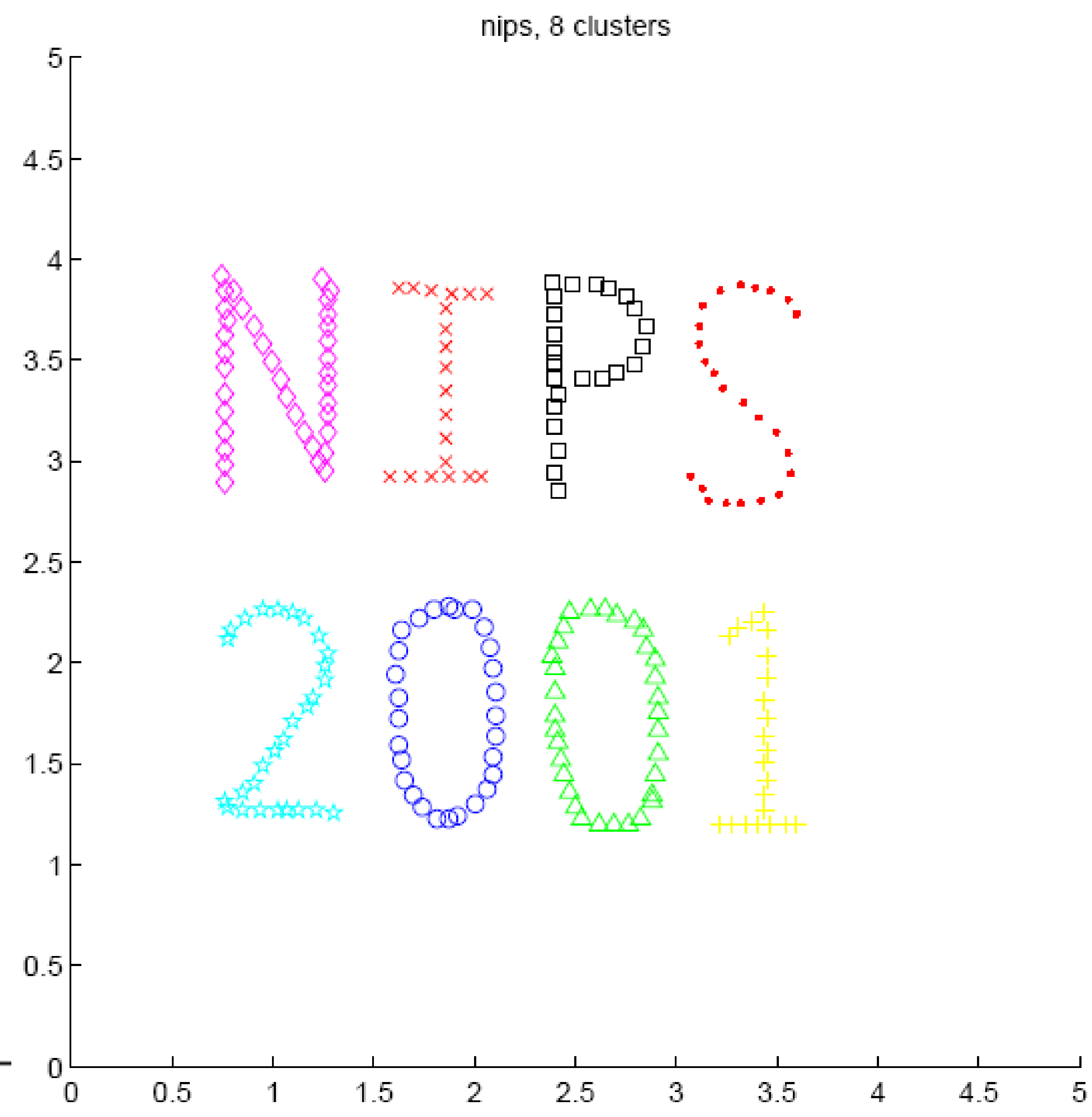
Second eigenvector of graph Laplacian



slide by Aarti Singh

Examples





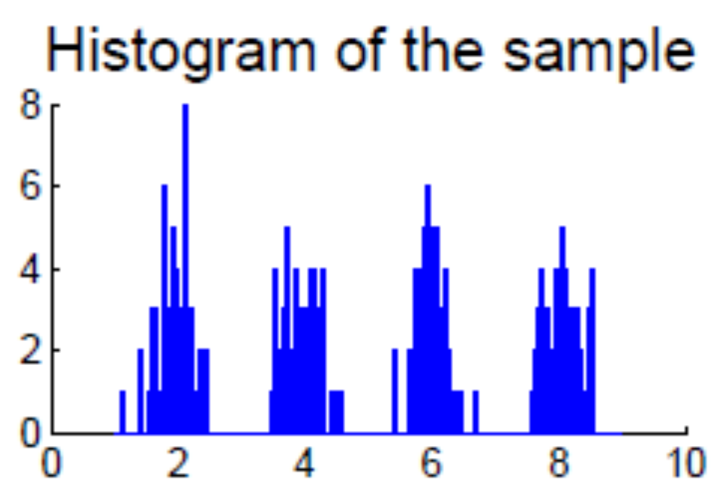
[Ng et al., 2001]

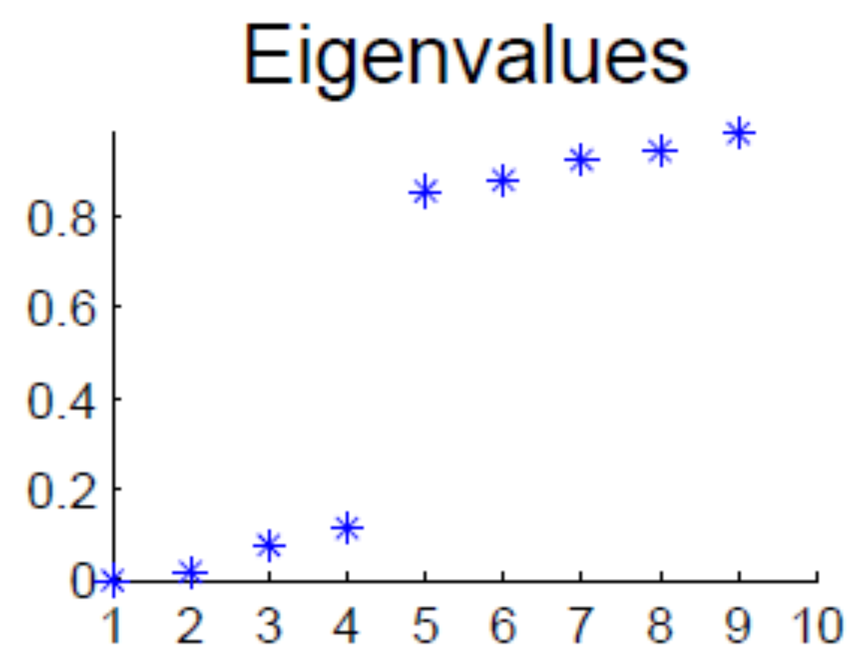
slide by Aarti Singh

Some Issues

- Choice of number of clusters k
  - Most stable clustering is usually given by the value of k that maximizes the eigengap (difference between consecutive eigenvalues)

$$\Delta_k = \left|\lambda - \lambda_{k-1}\right|$$



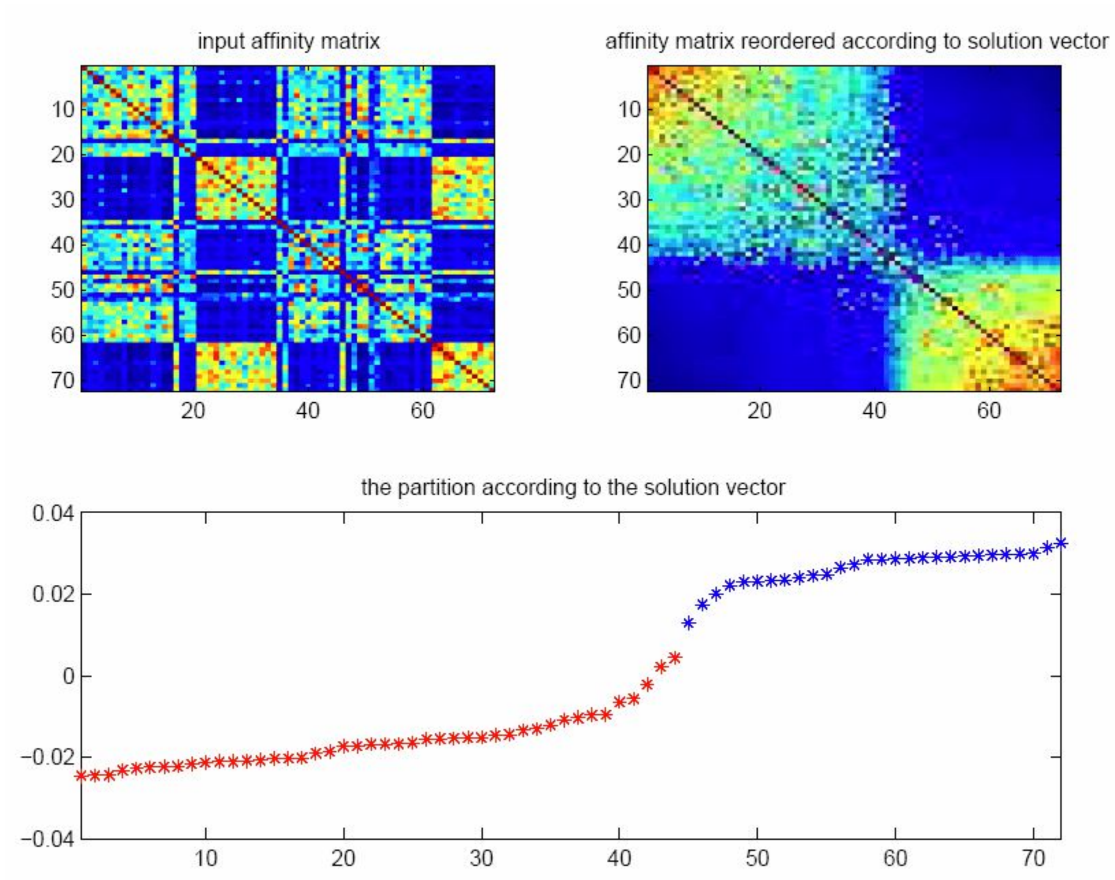


slide by Aarti Singh

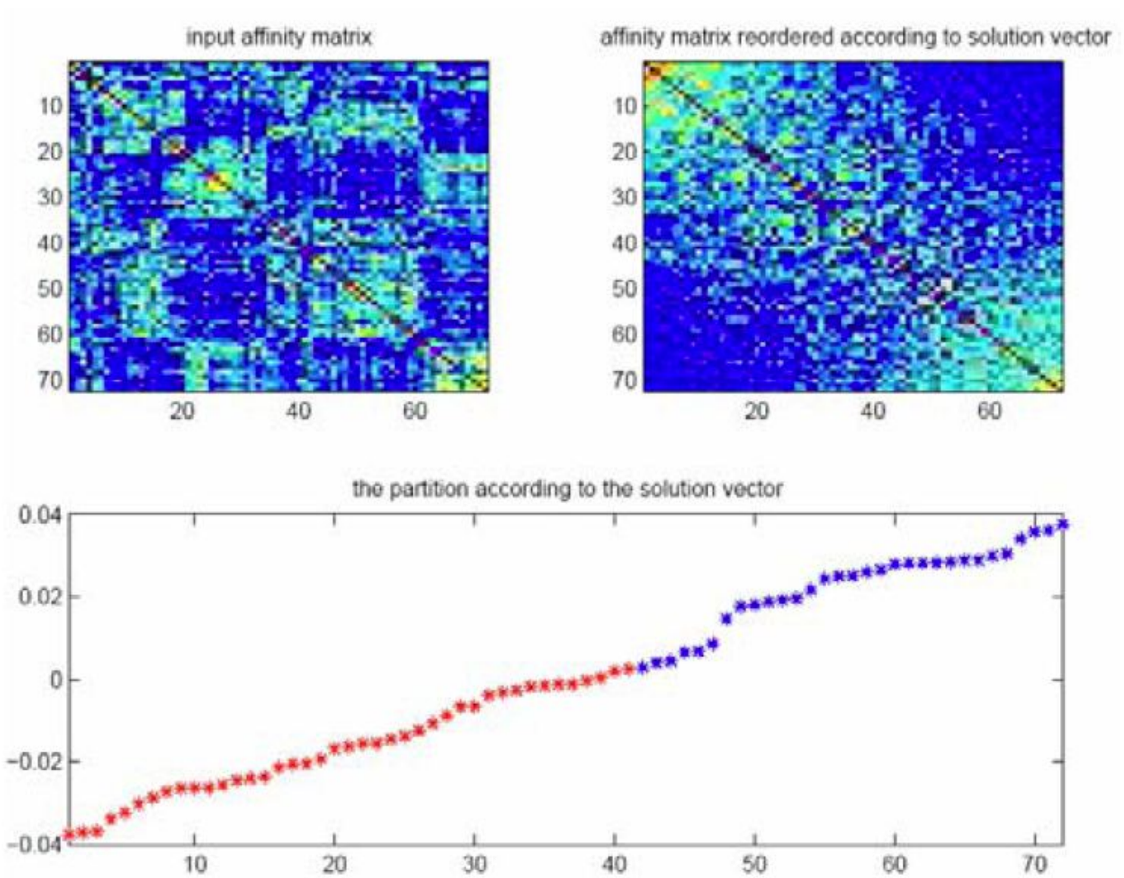
Some Issues

- Choice of number of clusters k
- Choice of similarity
  - Choice of kernel

    for Gaussian kernels, choice of σ



Good similarity measure



Poor similarity measure

slide by Aarti Singh

Some Issues

- Choice of number of clusters k

- Choice of similarity
  - Choice of kernel

  for Gaussian kernels, choice of σ

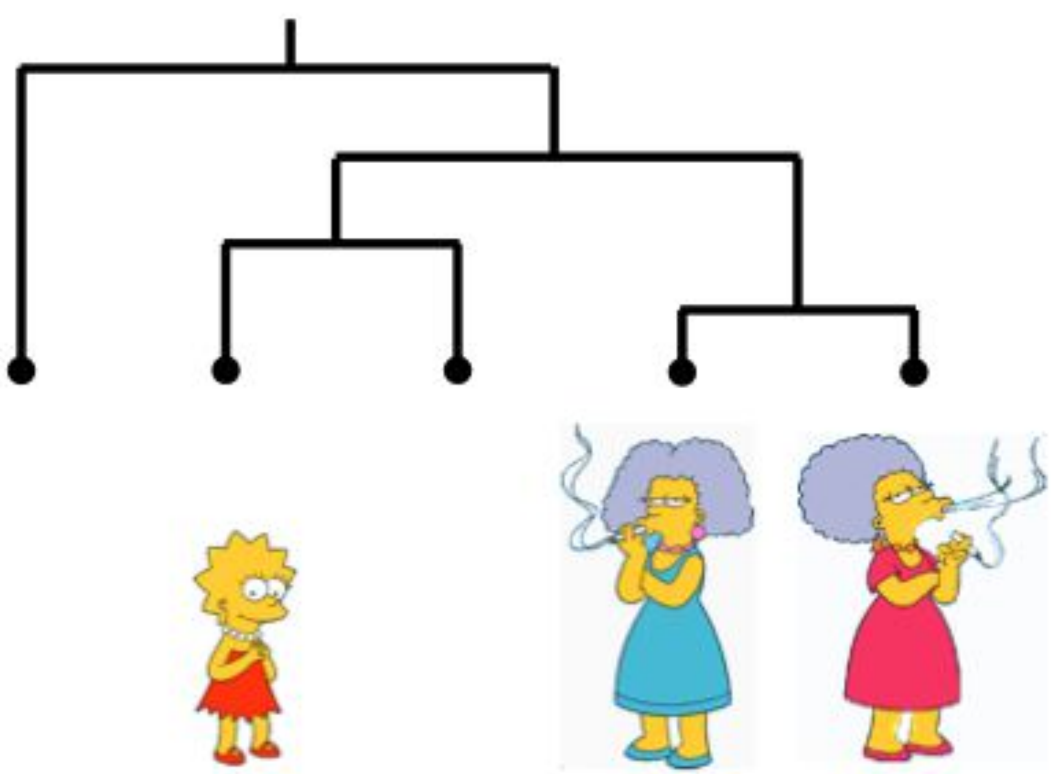
- Choice of clustering method
  - k-way vs. recursive 2-way

slide by Aarti Singh

Hierarchical clustering

Hierarchical Clustering

- Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



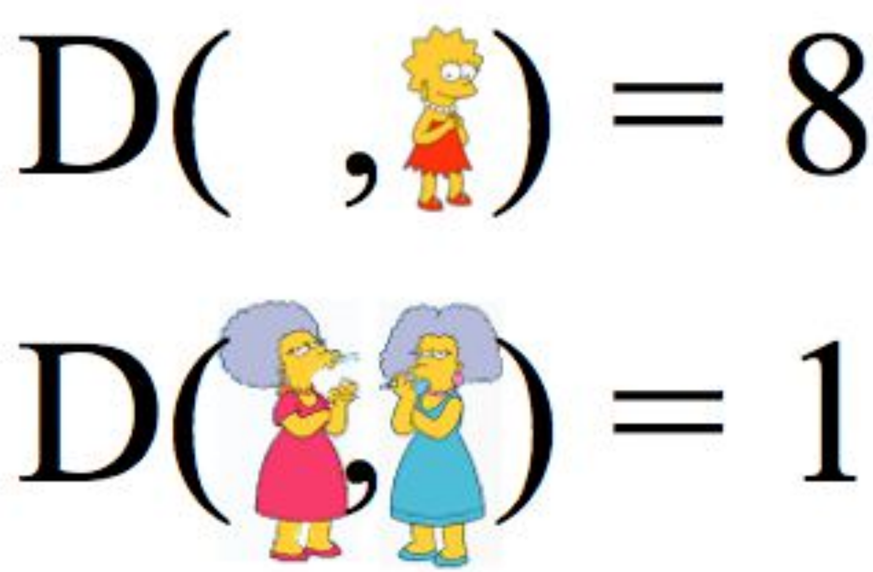
- The number of dendrograms with
  n leafs = (2n -3)!/[(2(n -2)) (n -2)!]

Number of leafs	Number of possible Dendrongrams
2	1
3	3
4	15
5	105
…	…
10	34,459,425

slide by Andrew Moore

We begin with a distance matrix which contains the distances between every pair of objects in our dataset

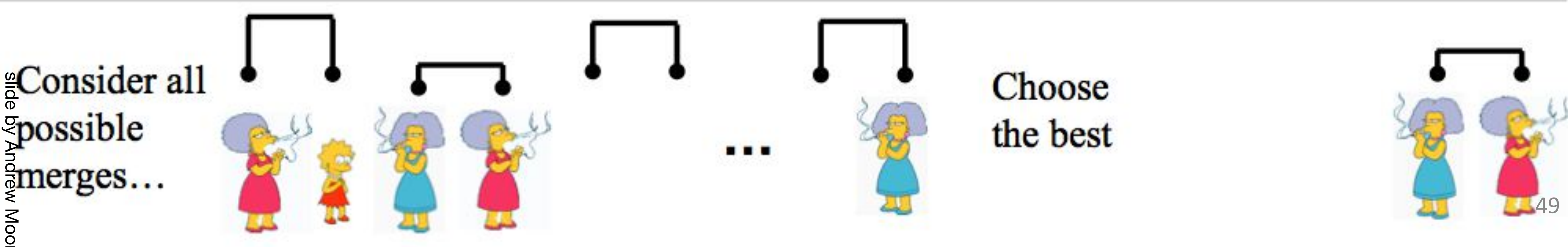
0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0



slide by Andrew Moore

Bottom-Up (agglomerative):

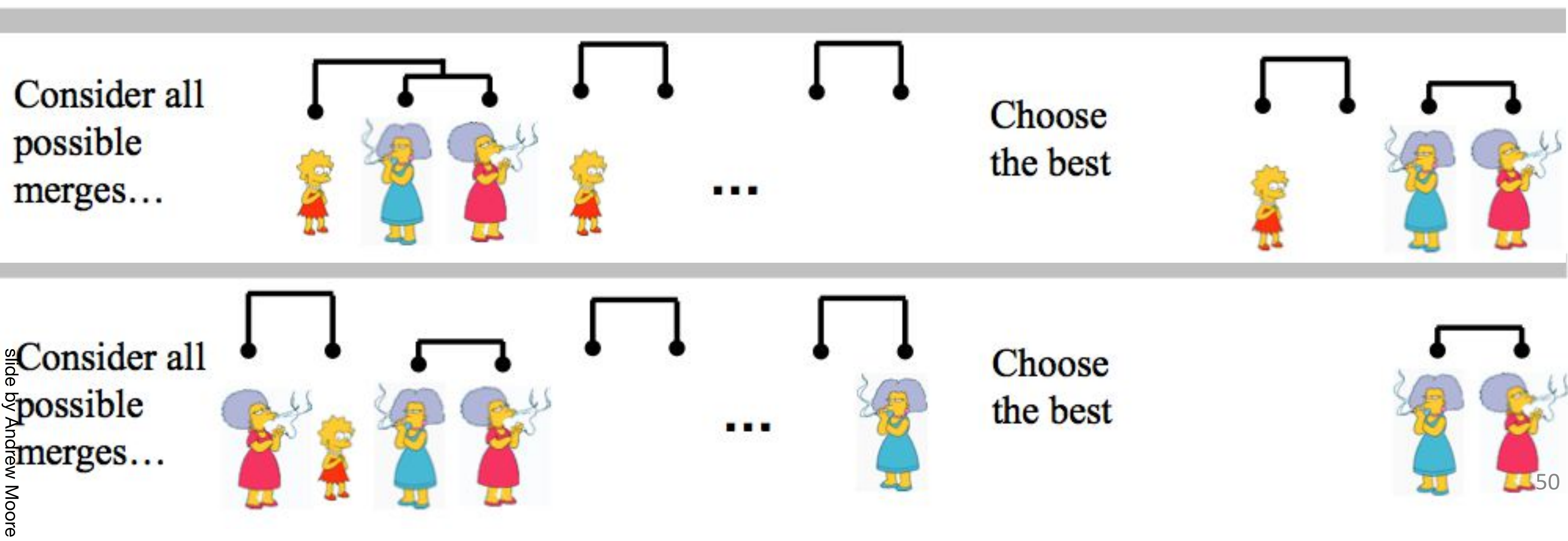
Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



slide by Andrew Moore

Bottom-Up (agglomerative):

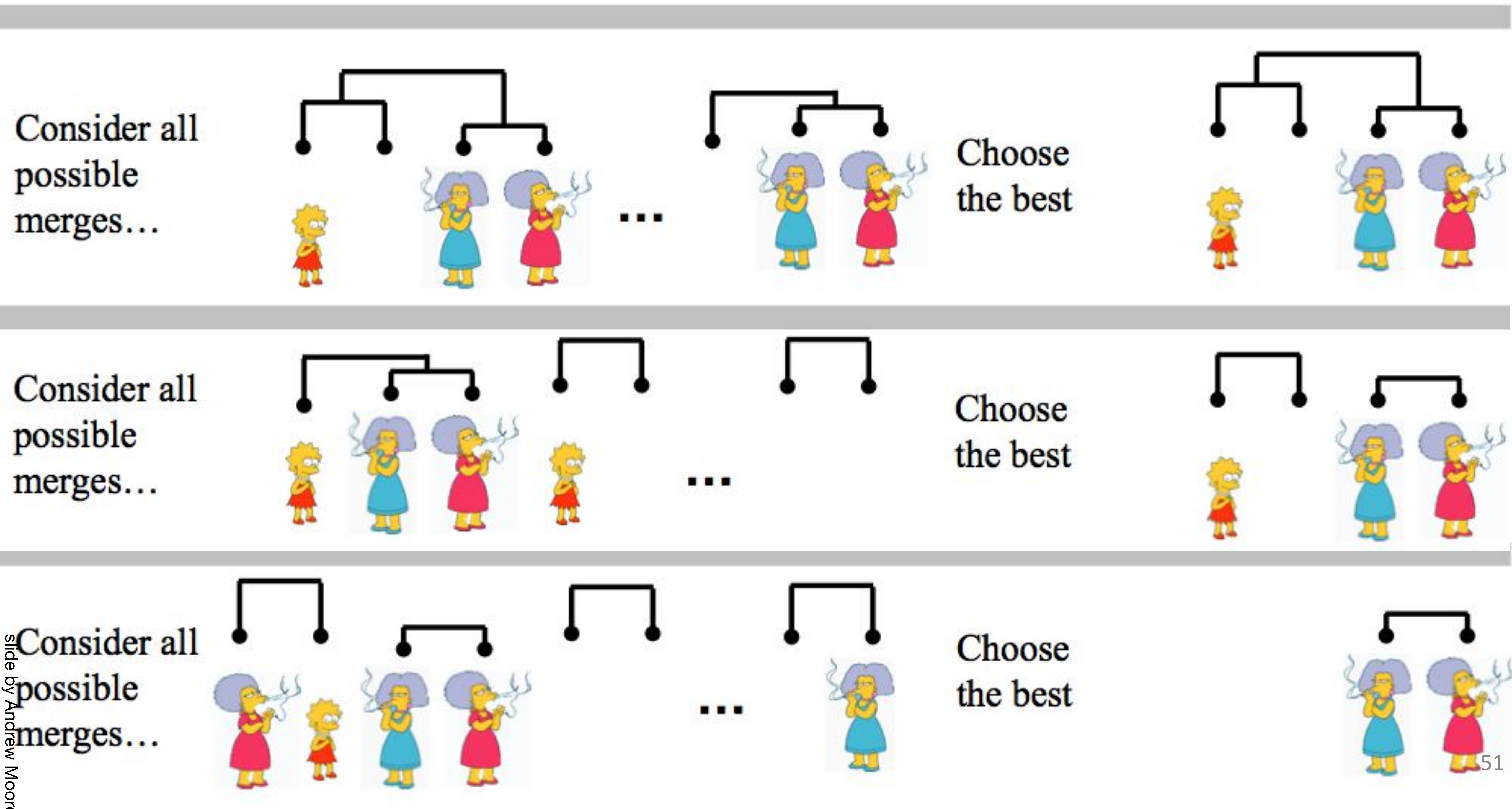
Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



slide by Andrew Moore

Bottom-Up (agglomerative):

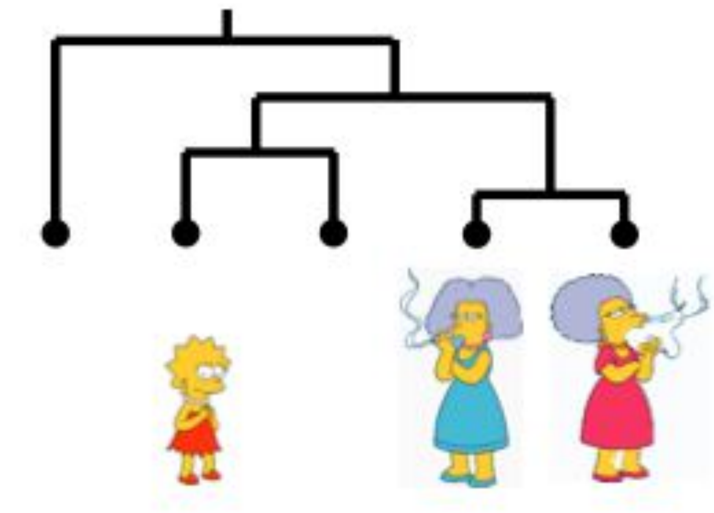
Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

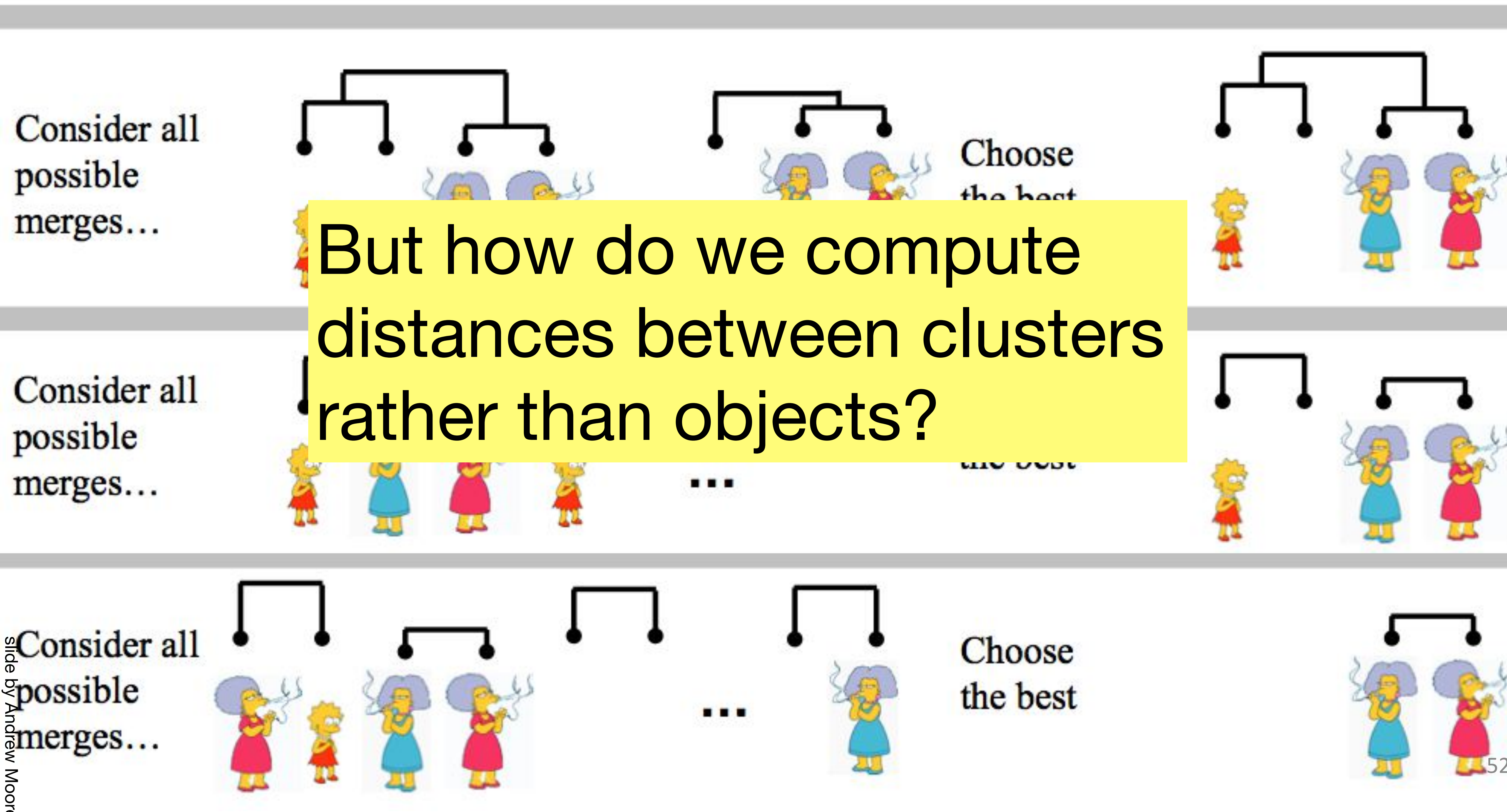


slide by Andrew Moore

Bottom-Up (agglomerative):

Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



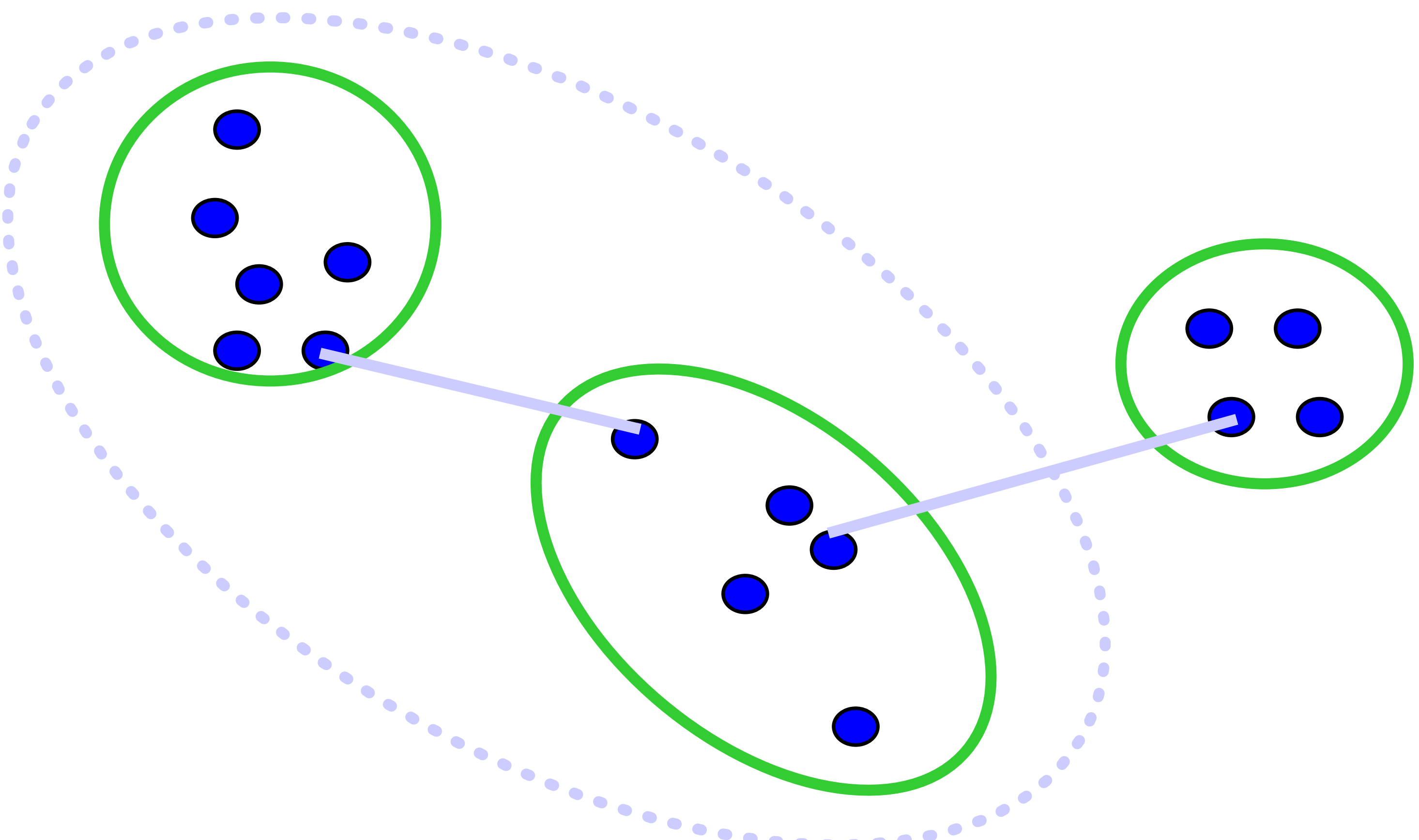


But how do we compute distances between clusters rather than objects?

slide by Andrew Moore

Computing distance between clusters: **Single Link**

- Cluster distance = distance of two closest members in each class

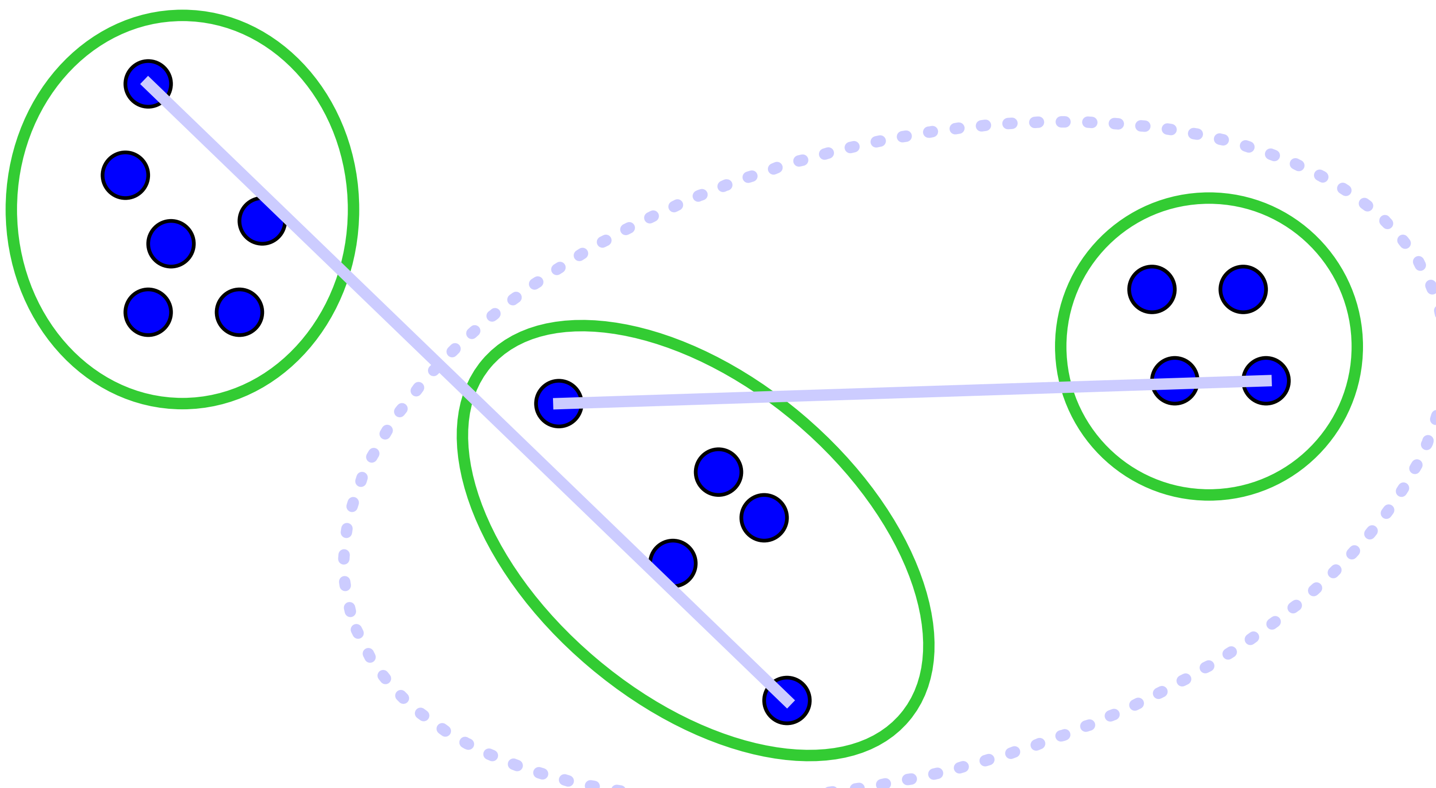


- Potentially long and skinny clusters

Computing distance between clusters: **Complete Link**

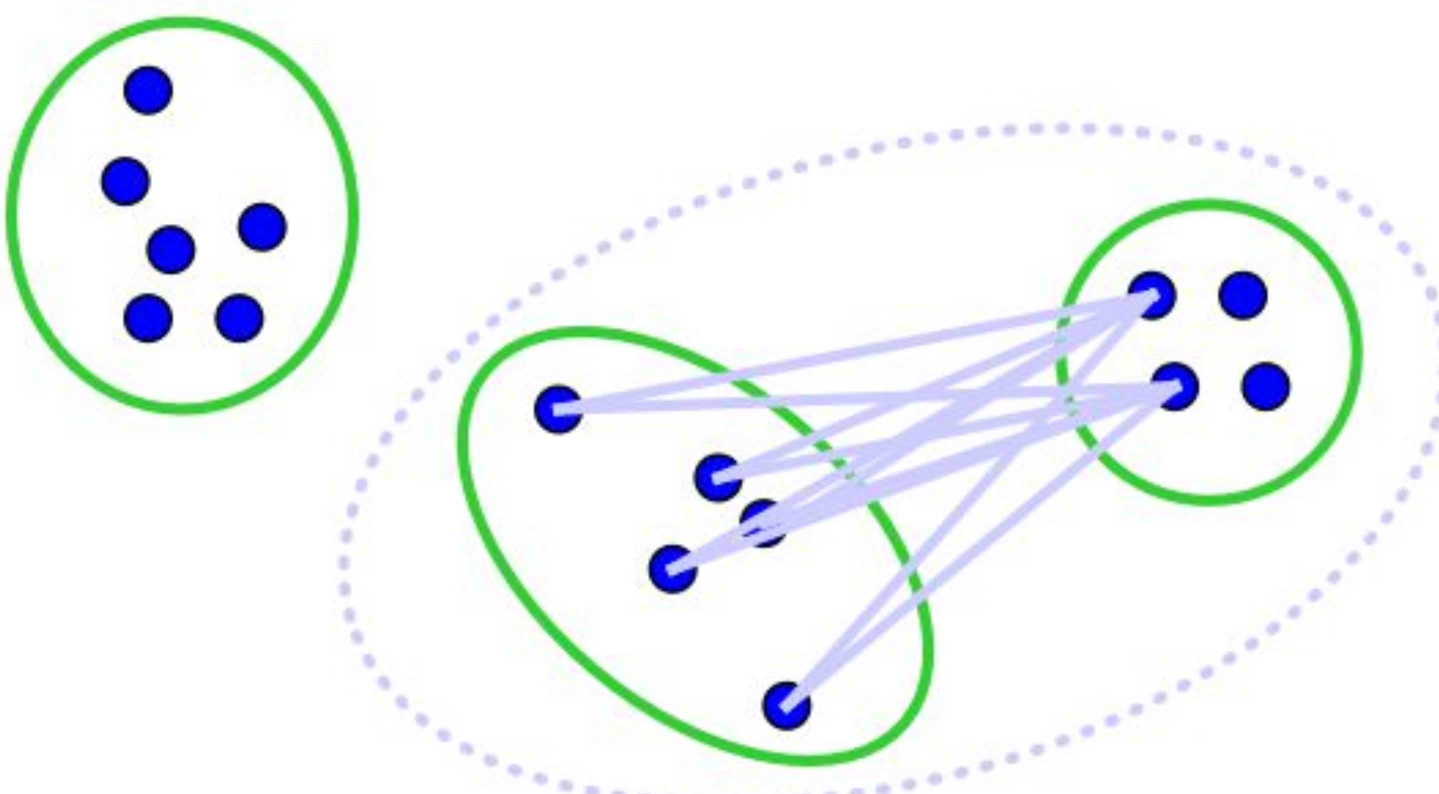
- Cluster distance = distance of two farthest members in each class

- Tight clusters



Computing distance between clusters: **Average Link**

- Cluster distance = average distance of all pairs



- The most widely used measure
- Robust against noise

slide by Andrew Moore

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
  - **silhouette coefficient**
- Need to use an “ultrametric” to get a meaningful hierarchy

slide by Derek Hoiem