Bag of features

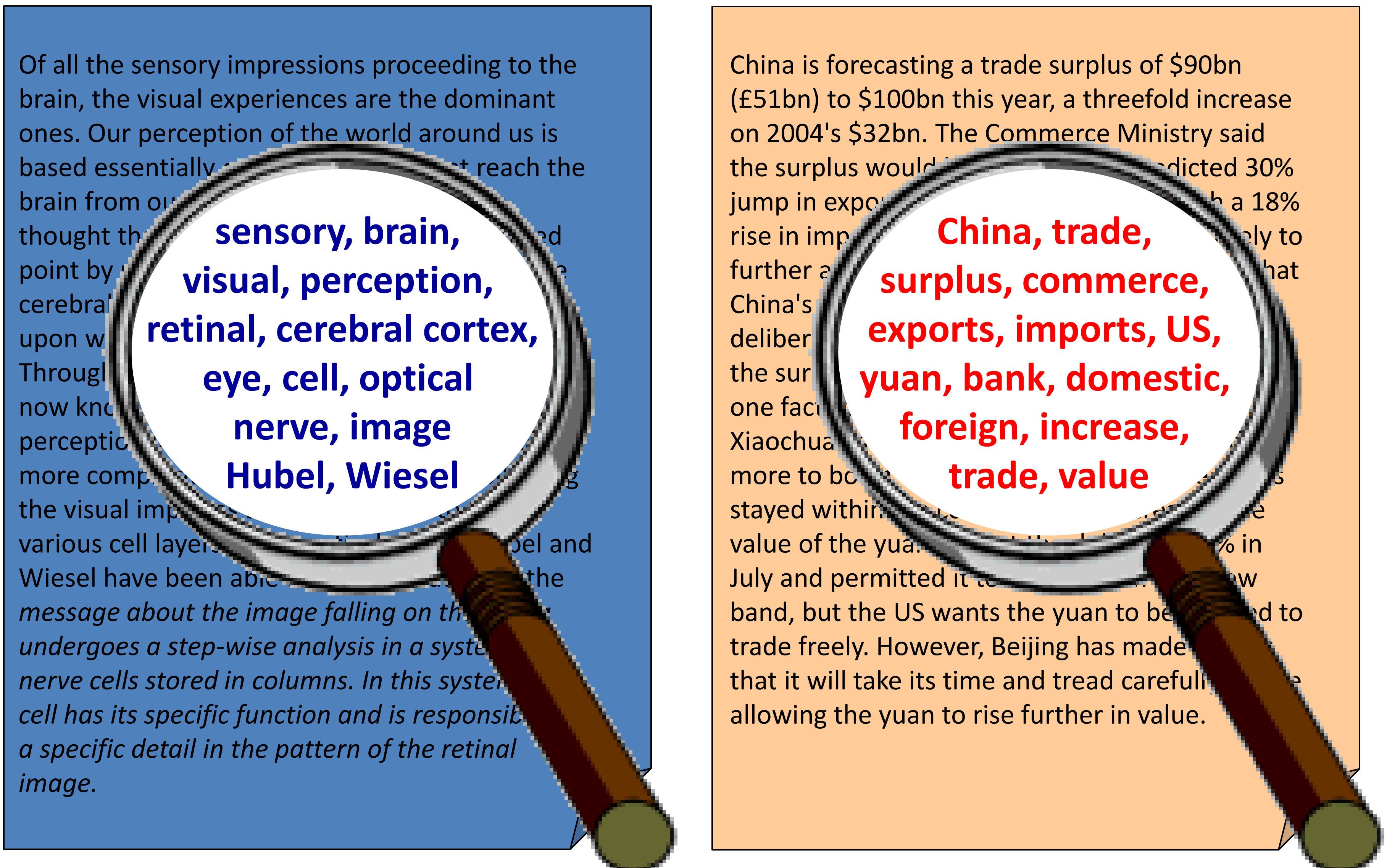
Gonzalo Vaca-Castano

REU 2013

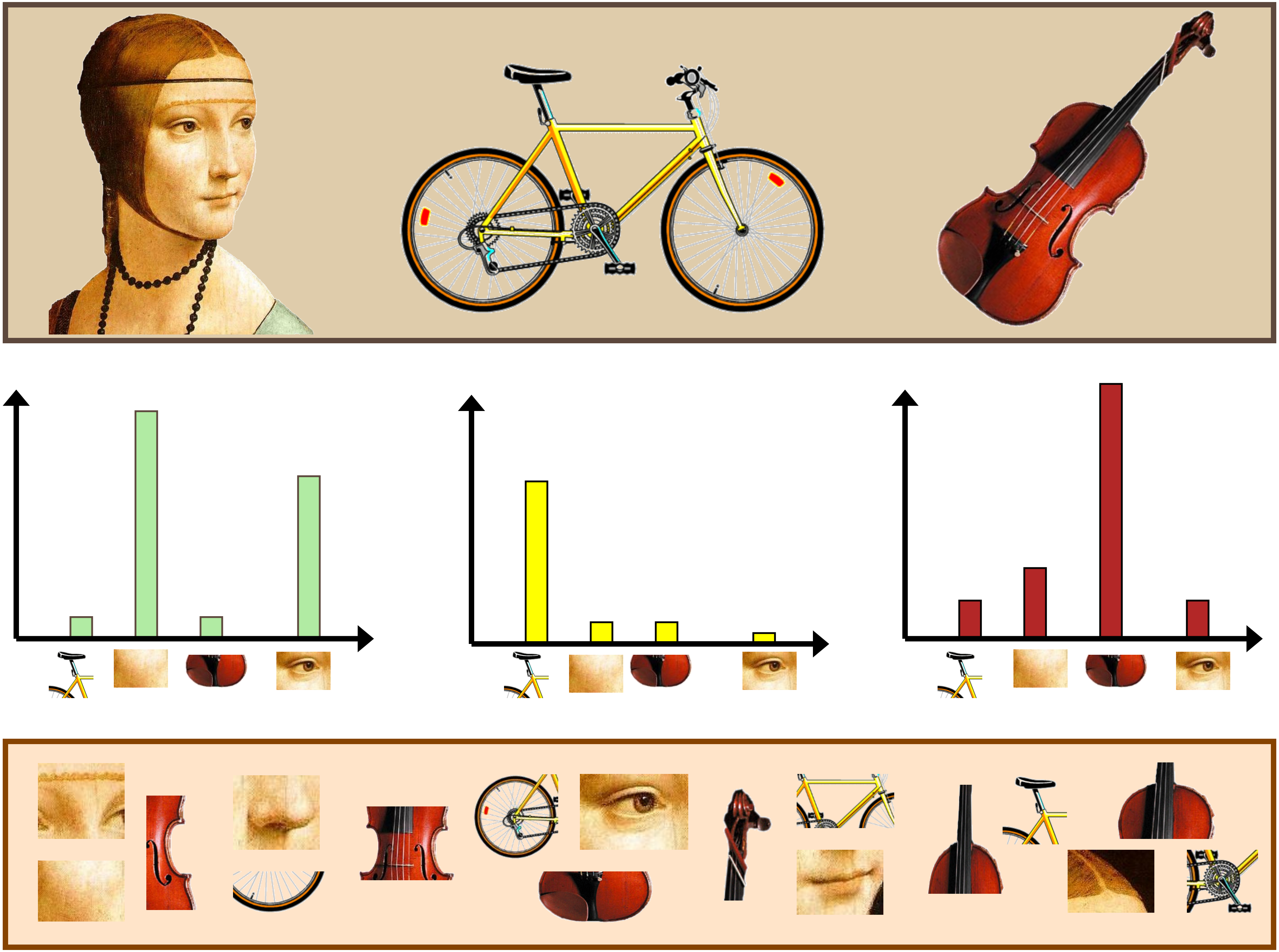
Object → **Bag of 'words'**

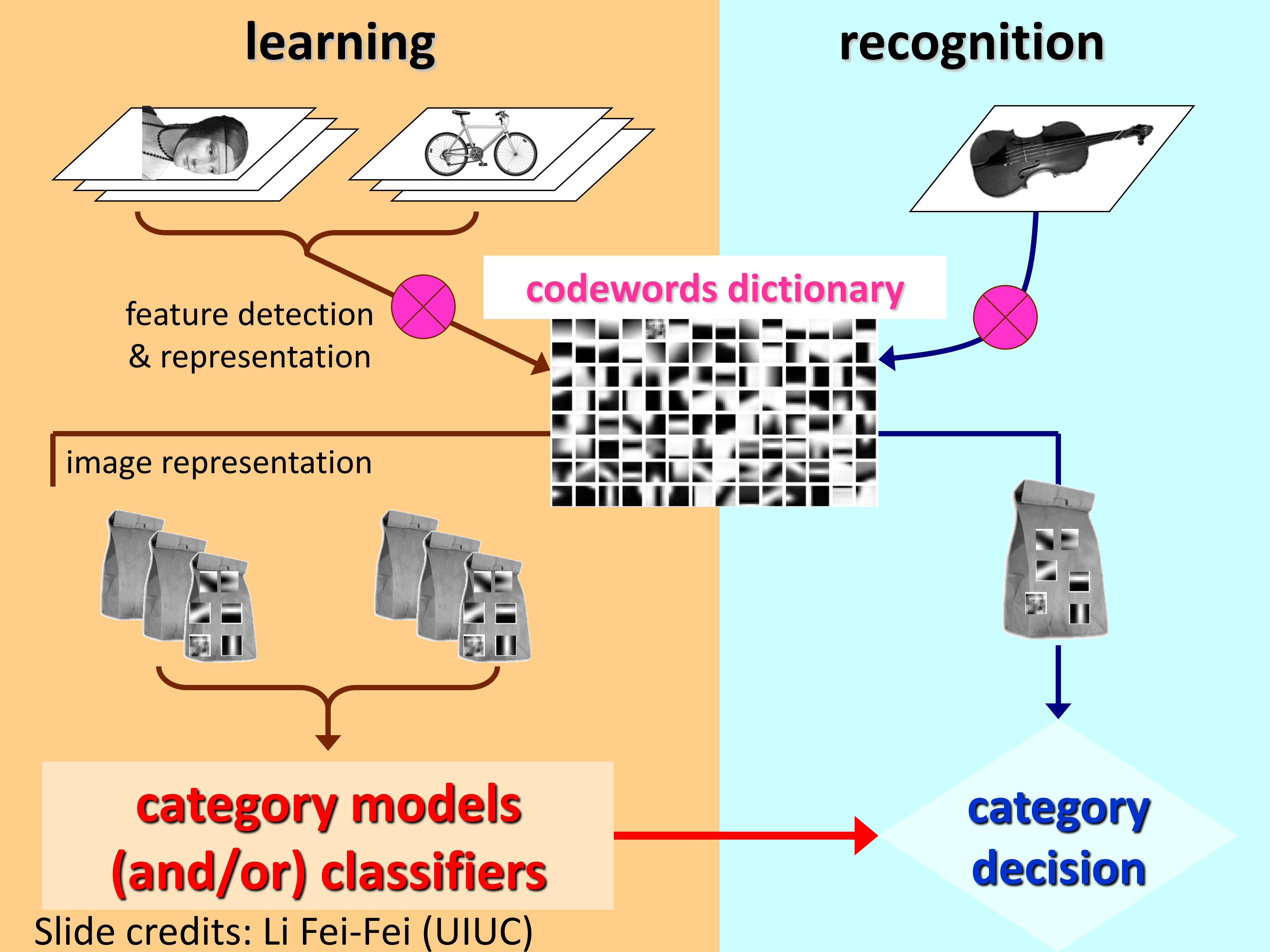
Slide credits: Li Fei-Fei (UIUC)

Analogy to documents

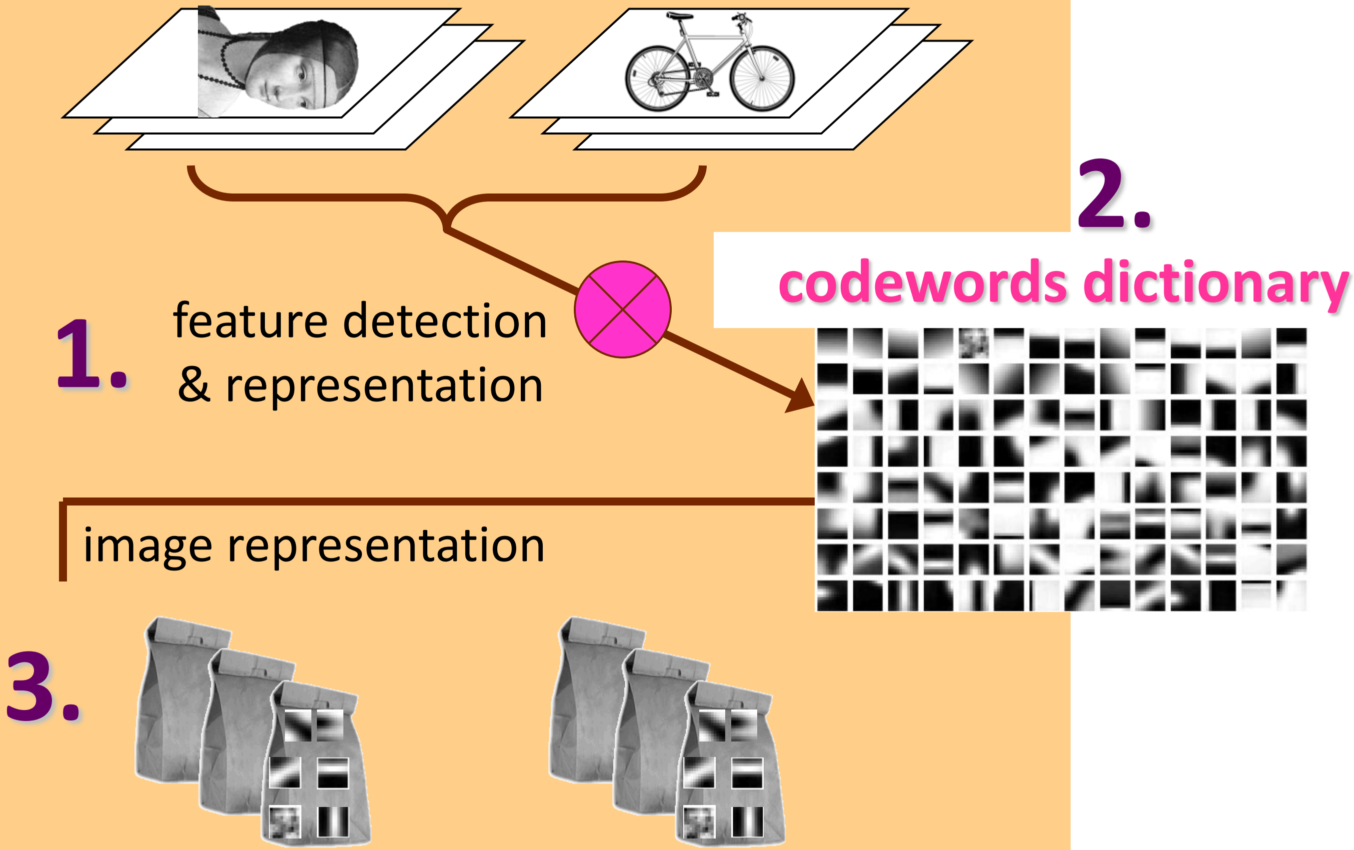


Slide credits: Li Fei-Fei (UIUC)



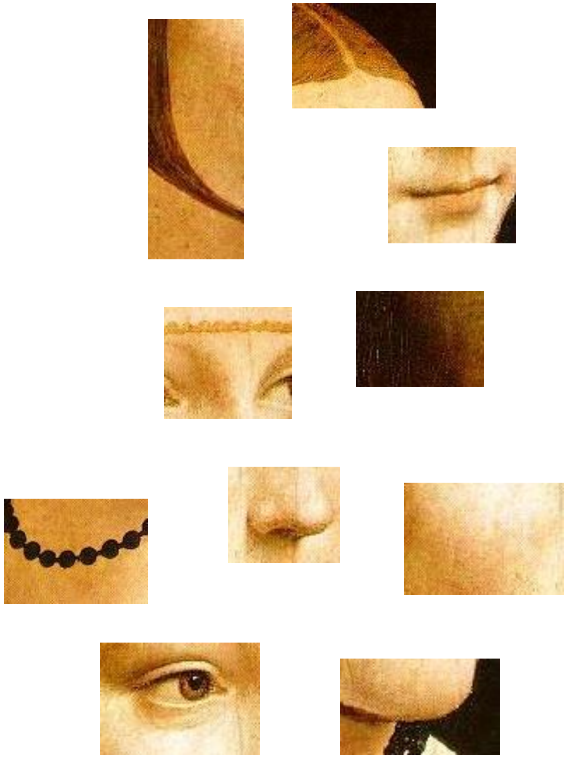

learning
recognition
feature detection
& representation
codewords dictionary
image representation
category models
(and/or) classifiers
category
decision
Slide credits: Li Fei-Fei (UIUC)

Representation



Slide credits: Li Fei-Fei (UIUC)

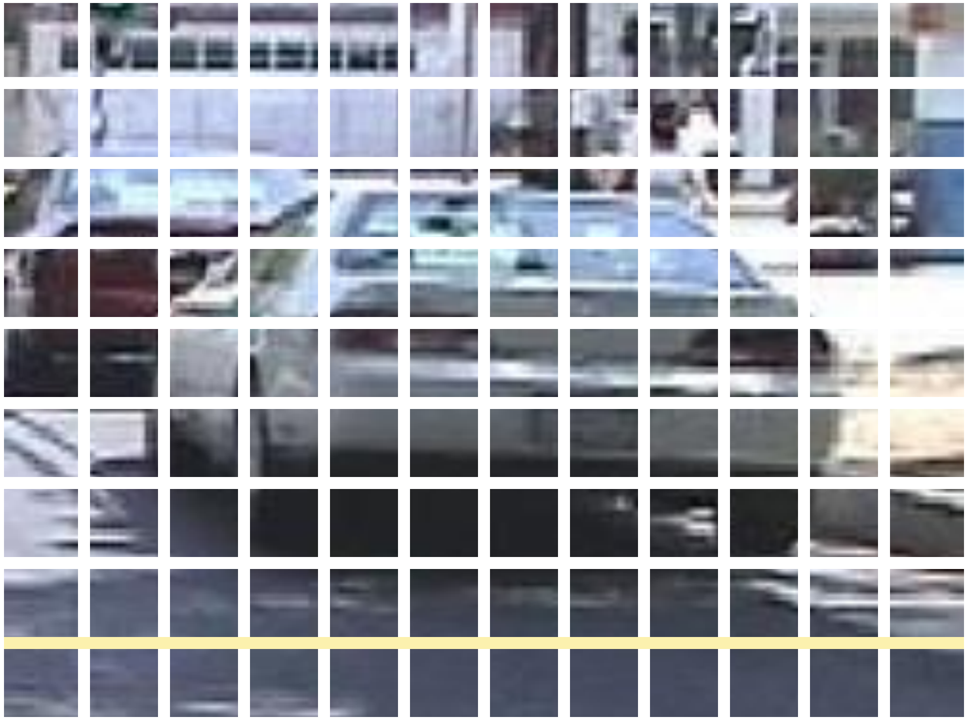
1.Feature detection and representation



Slide credits: Li Fei-Fei (UIUC)

1.Feature detection and representation

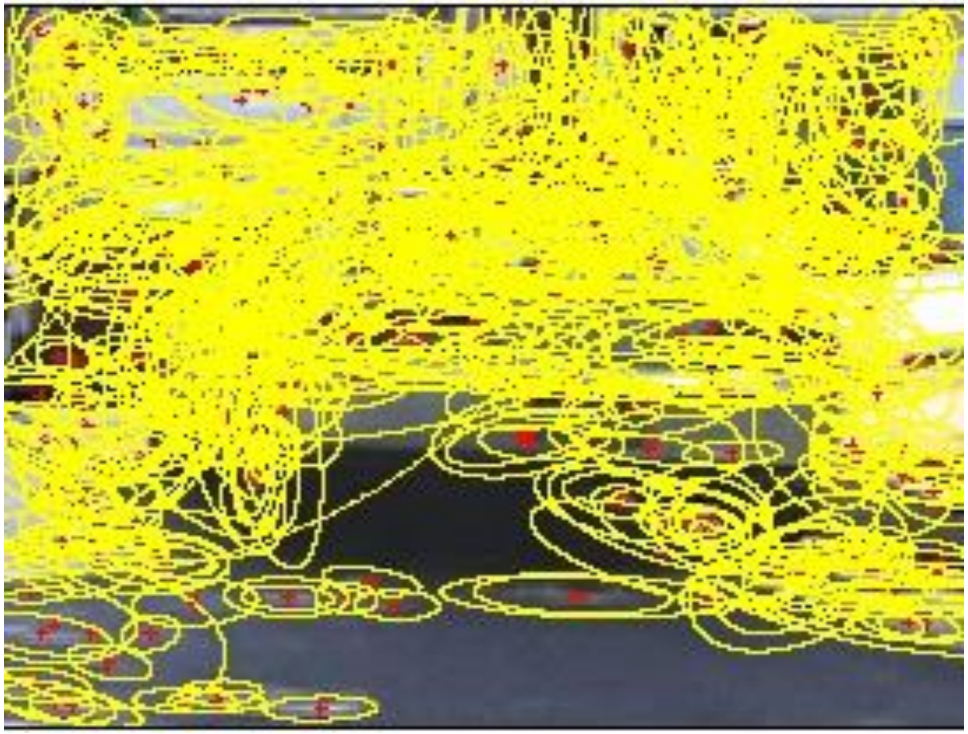
- Regular grid
  - Vogel et al. 2003
  - Fei-Fei et al. 2005



Slide credit: Josef Sivic

1.Feature detection and representation

- Regular grid
  - Vogel et al. 2003
  - Fei-Fei et al. 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei et al. 2005
  - Sivic et al. 2005



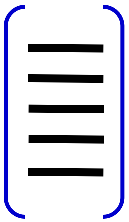
Slide credit: Josef Sivic

1.Feature detection and representation

- Regular grid
  - Vogel et al. 2003
  - Fei-Fei et al. 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei et al. 2005
  - Sivic et al. 2005
- Other methods
  - Random sampling (Ullman et al. 2002)
  - Segmentation based patches (Barnard et al. 2003)

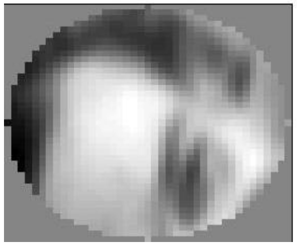
Slide credit: Josef Sivic

1.Feature detection and representation

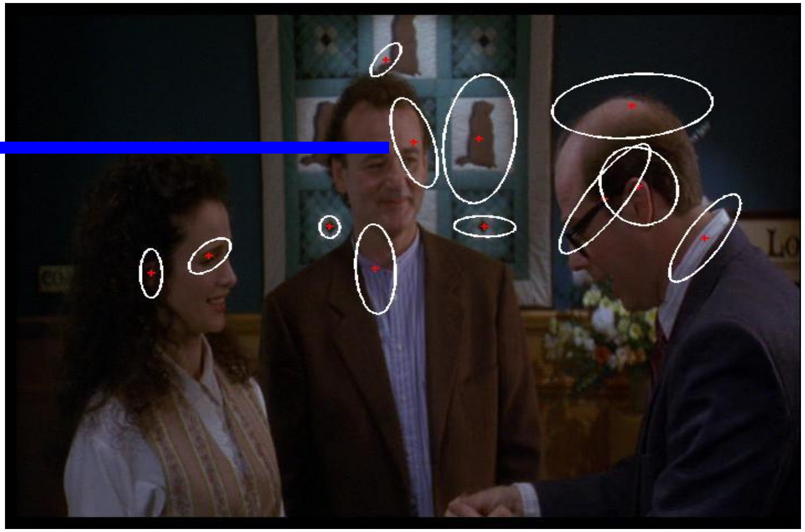


Compute SIFT descriptor

[Lowe’99]



Normalize patch



Detect patches

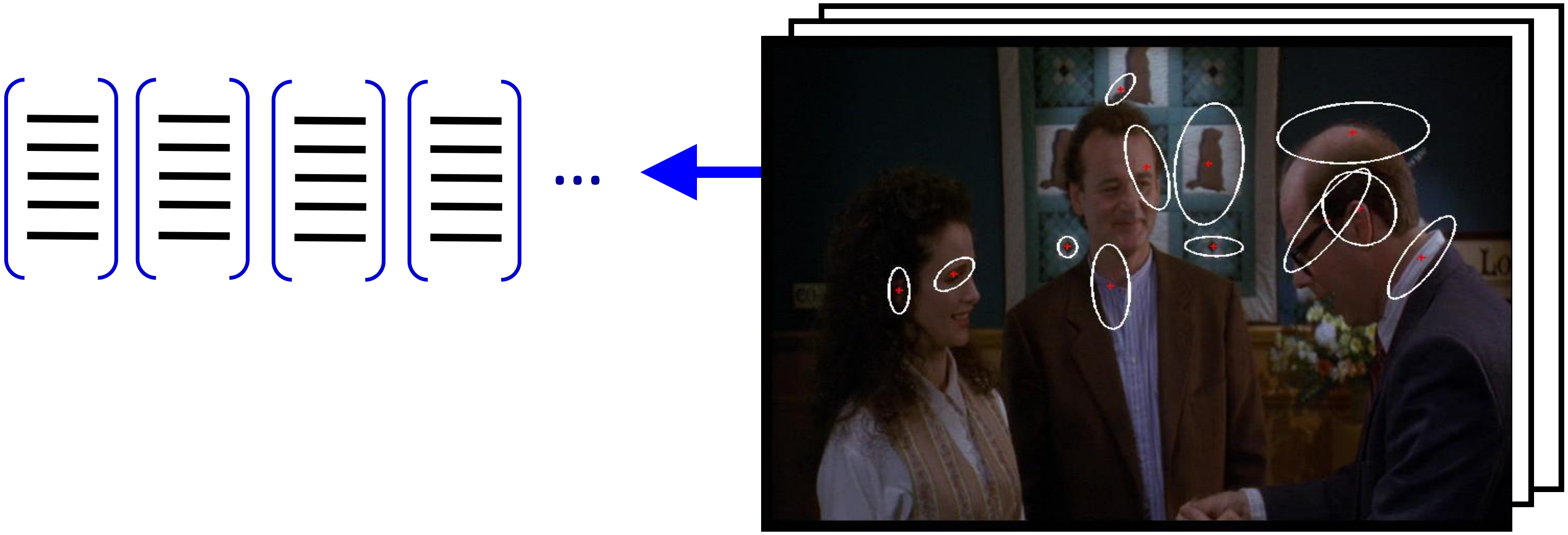
[Mikojaczyk and Schmid ’02]

[Matas et al. ’02]

[Sivic et al. ’03]

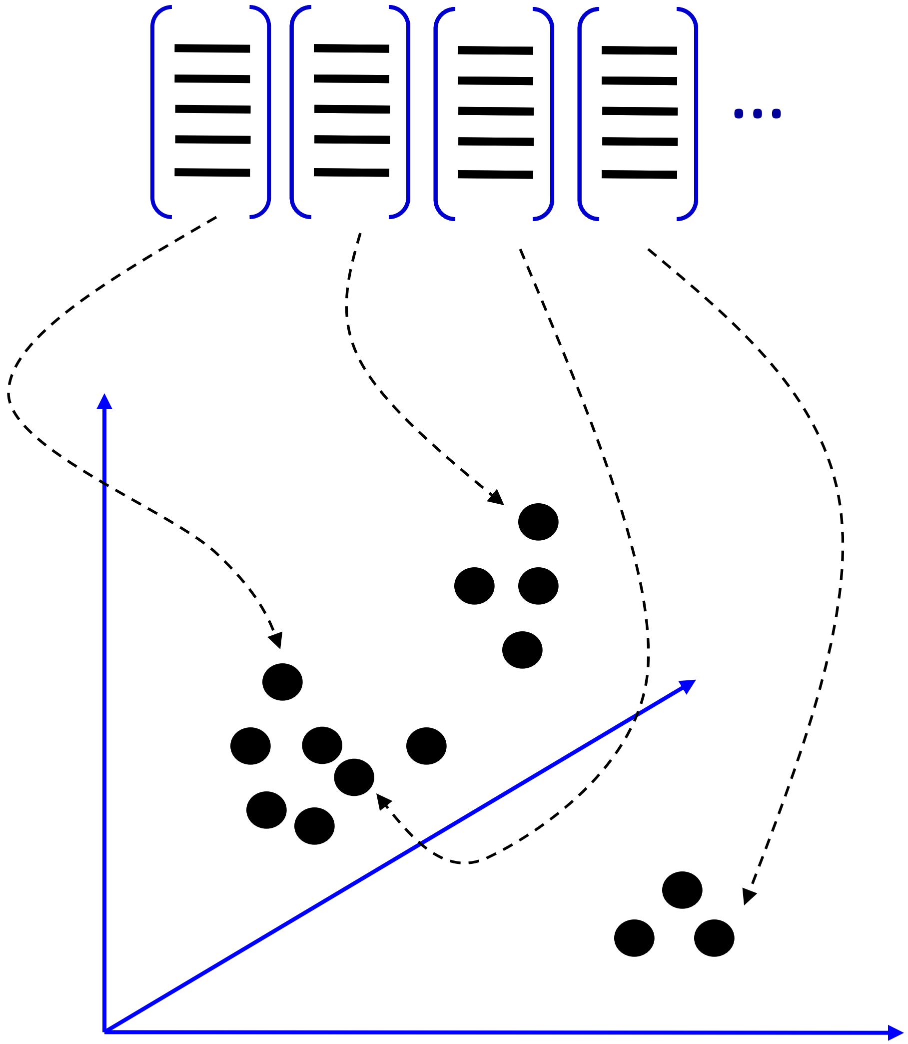
Slide credit: Josef Sivic

1.Feature detection and representation



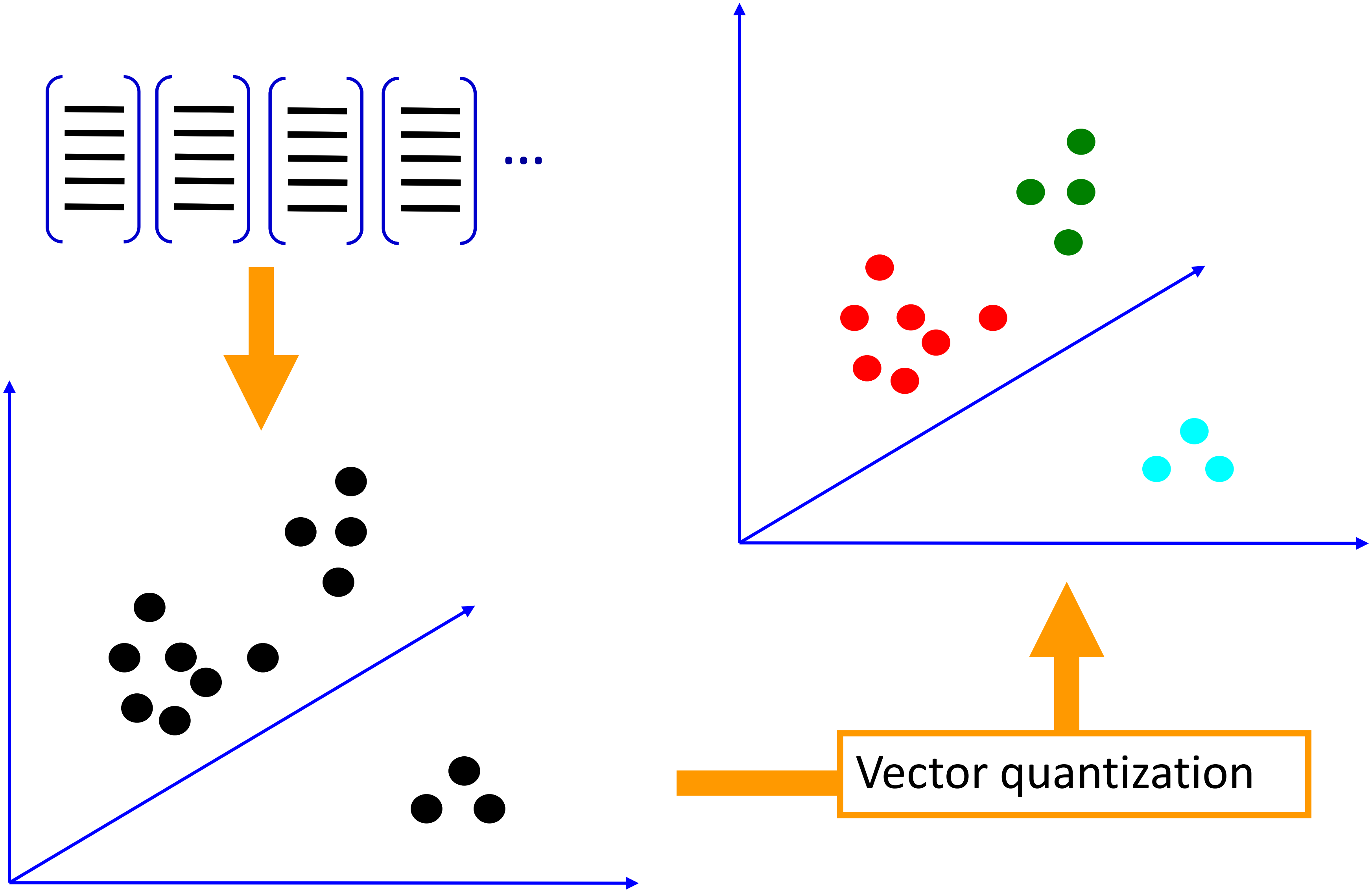
Slide credit: Josef Sivic

2. Codewords dictionary formation



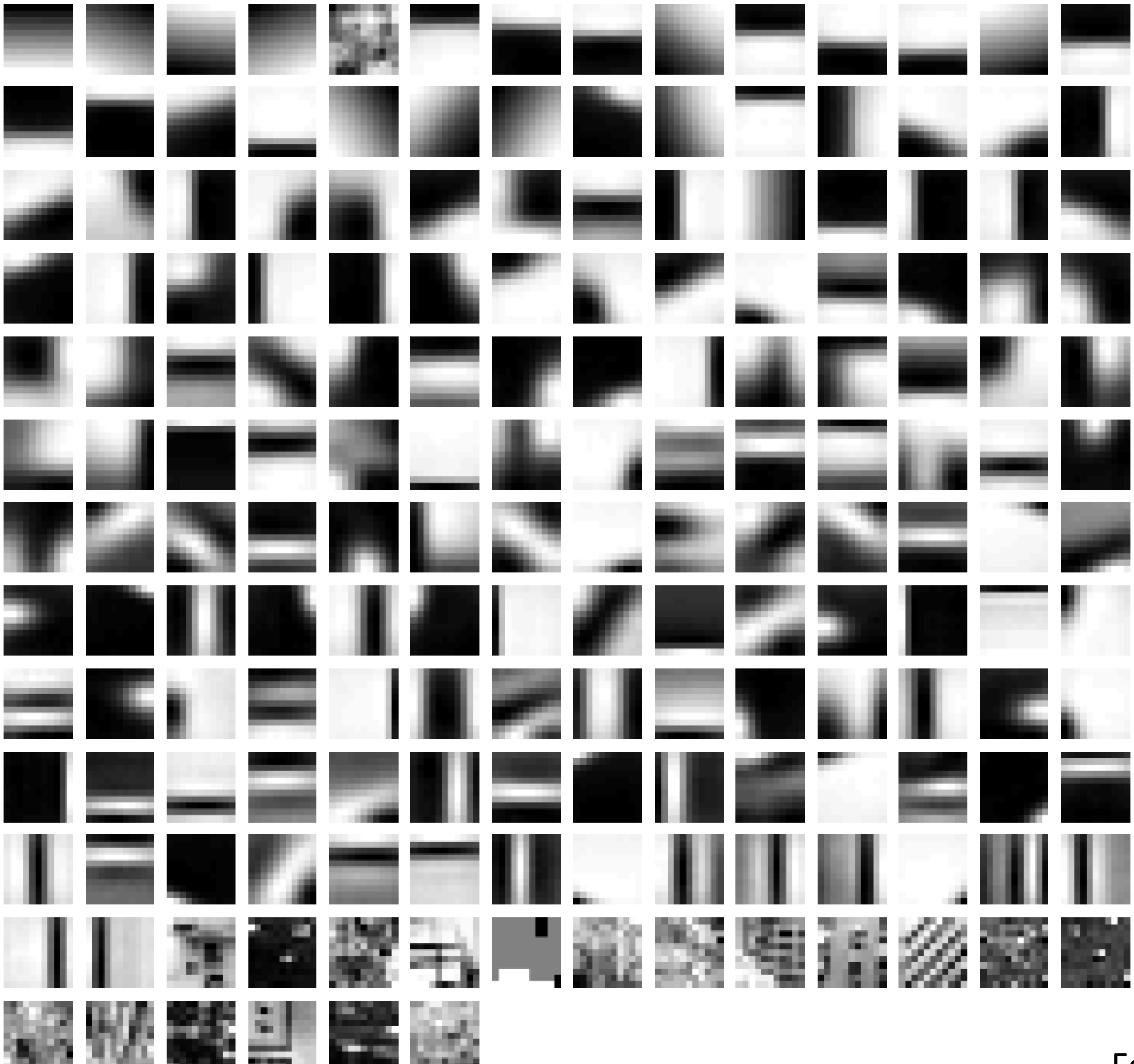
Slide credit: Josef Sivic

2. Codewords dictionary formation



Slide credit: Josef Sivic

2. Codewords dictionary formation



Fei-Fei et al. 2005

How to create a dictionary ?

- Answer: Clustering
- Typical Algorithm: Kmeans
- Research Topic:
  - Clustering in High Dimensional data

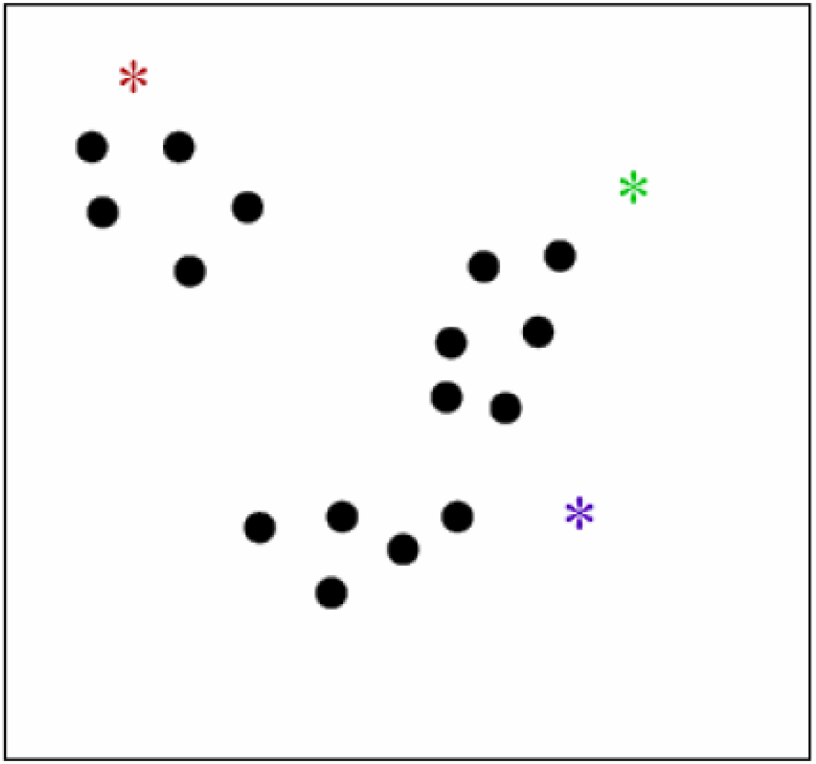
K-Means Algorithm

- K = # of clusters (given); one "mean" per cluster
- Interval data

- Initialize means (e.g. by picking k samples at random)
- Iterate:

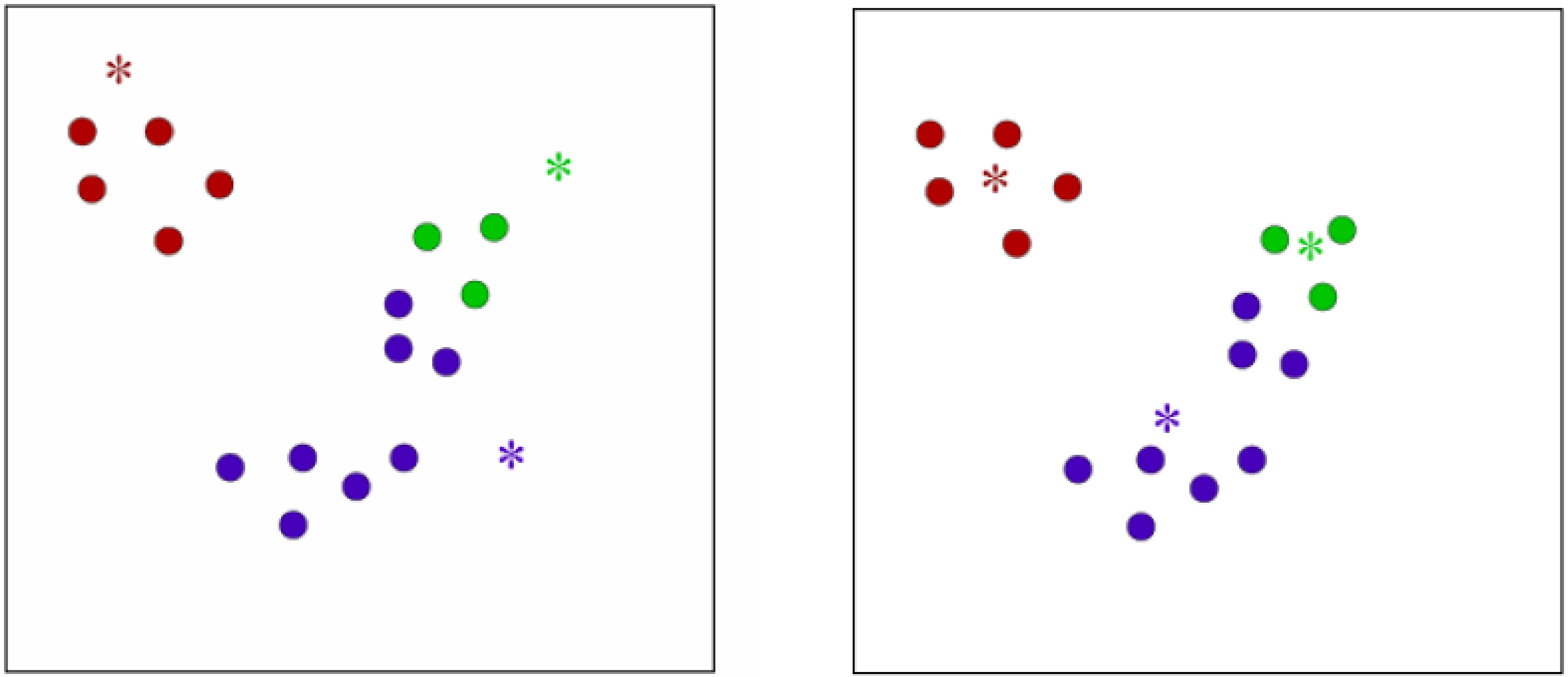
(1) assign each point to nearest mean

(2) move "mean" to center of its cluster.



Initialize representatives ("means")

Assignment Step; Means Update



Assign to nearest representative | Re-estimate means

Bregman Hard Clustering

- Initialize $\{\mu_h\}_{h=1}^{k}$
- Repeat until *convergence*
  - { Assignment Step }
    Assign x to $\mathcal{X}_h$ if $h = \underset{h'}{\mathrm{argmin}}\ d_\phi(\mathrm{x}, \mu_{h'})$
  - { Re-estimation step }
    For all h

$$\mu_h = \frac{\sum_{\mathrm{x} \in \mathcal{X}_h} p(\mathrm{x})\, \mathrm{x}}{\sum_{\mathrm{x} \in \mathcal{X}_h} p(\mathrm{x})}$$

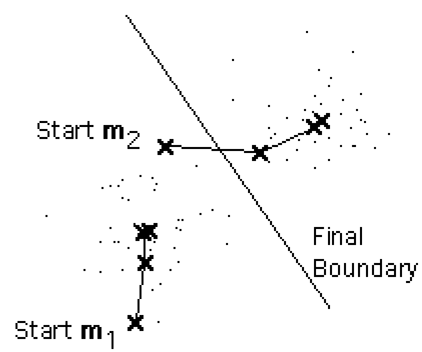
1. *Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.*
2. *Assign each object to the group that has the closest centroid.*
3. *When all objects have been assigned, recalculate the positions of the K centroids.*
4. *Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.*

An example

Suppose that we have n sample feature vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ all from the same class, and we know that they fall into k compact clusters, k < n. Let $\mathbf{m}_i$ be the mean of the vectors in cluster i. If the clusters are well separated, we can use a minimum-distance classifier to separate them. That is, we can say that $\mathbf{x}$ is in cluster i if $\| \mathbf{x} - \mathbf{m}_i \|$ is the minimum of all the k distances. This suggests the following procedure for finding the k means:

- Make initial guesses for the means $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_k$
- Until there are no changes in any mean
  - Use the estimated means to classify the samples into clusters
  - For i from 1 to k
    - Replace $\mathbf{m}_i$ with the mean of all of the samples for cluster i
  - end_for
- end_until

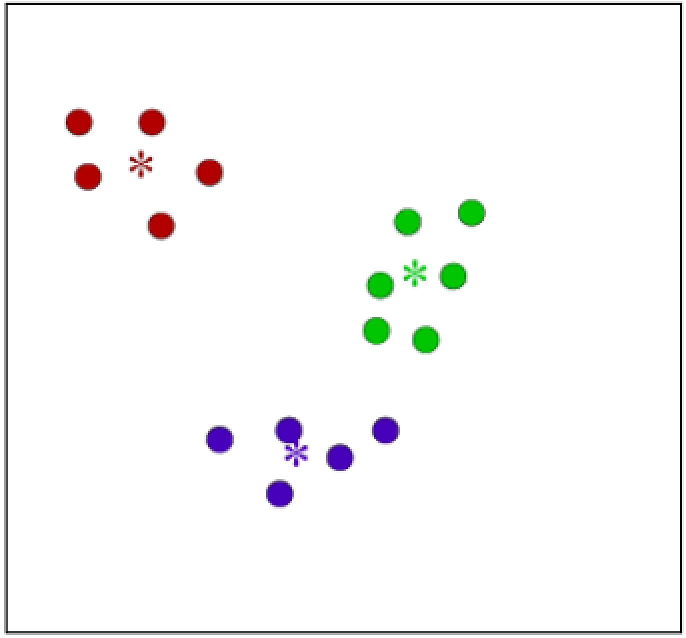
Here is an example showing how the means $\mathbf{m}_1$ and $\mathbf{m}_2$ move into the centers of two clusters.



Convergence after another iteration

Complexity:

O(k . n . # of iterations



The objective function is

$$\min_{\{\boldsymbol{\mu}_1,\cdots,\boldsymbol{\mu}_k\}} \sum_{h=1} \sum_{\mathbf{x}\in\mathcal{X}_h} \|\mathbf{x}-\boldsymbol{\mu}_h\|^2$$

K-means Clustering – Details

- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes
  - Easily parallelized
  - Use kd-trees or other efficient spatial data structures for some situations
    - Pelleg and Moore (X-means)

- Sensitivity to initial conditions

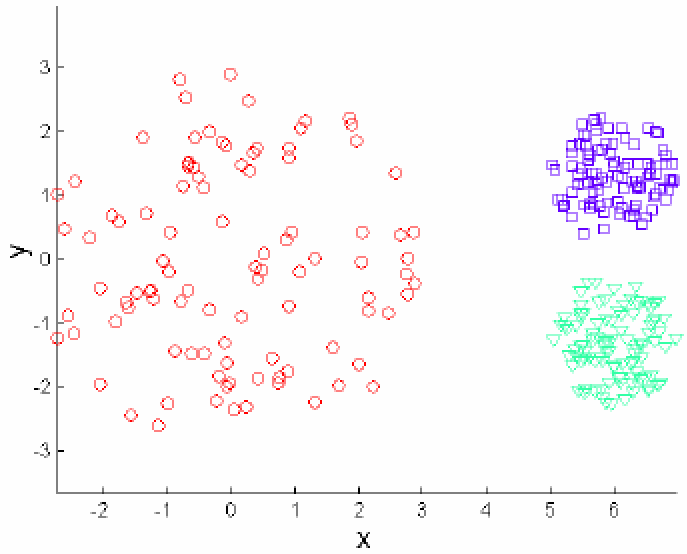
- A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

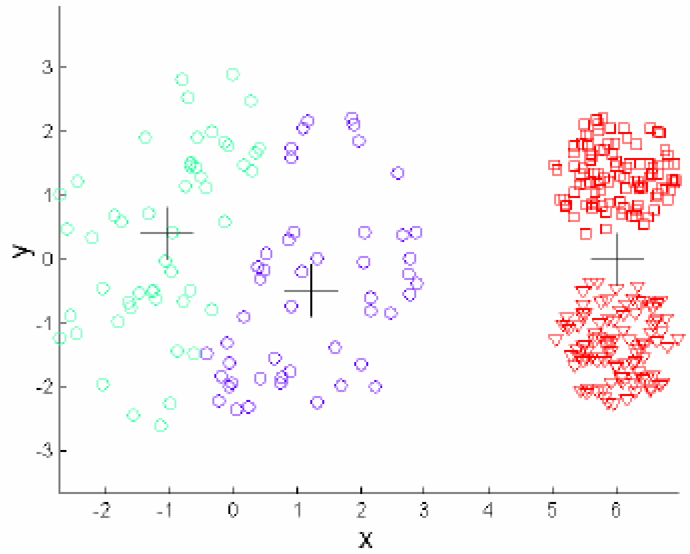
Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- Problems with outliers
- Empty clusters

Limitations of K-means: Differing Density

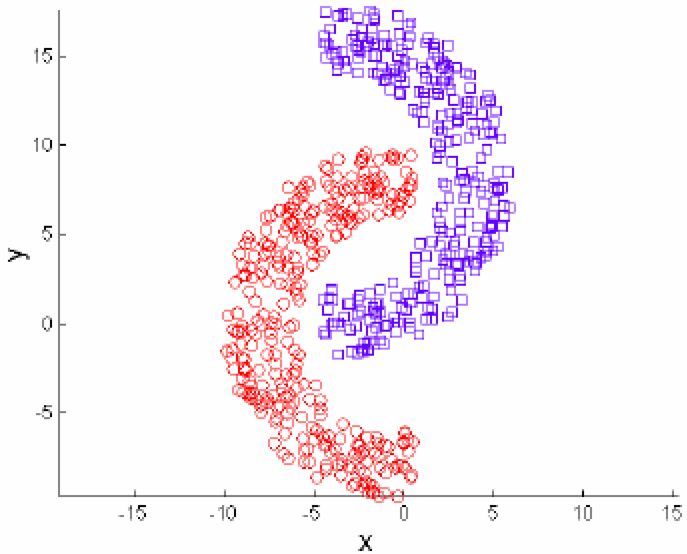




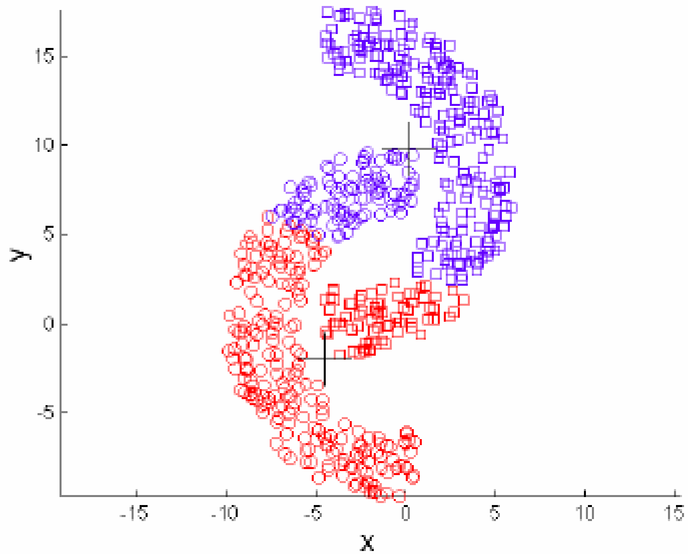
Original Points

K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points



K-means (2 Clusters)

K-mean Research

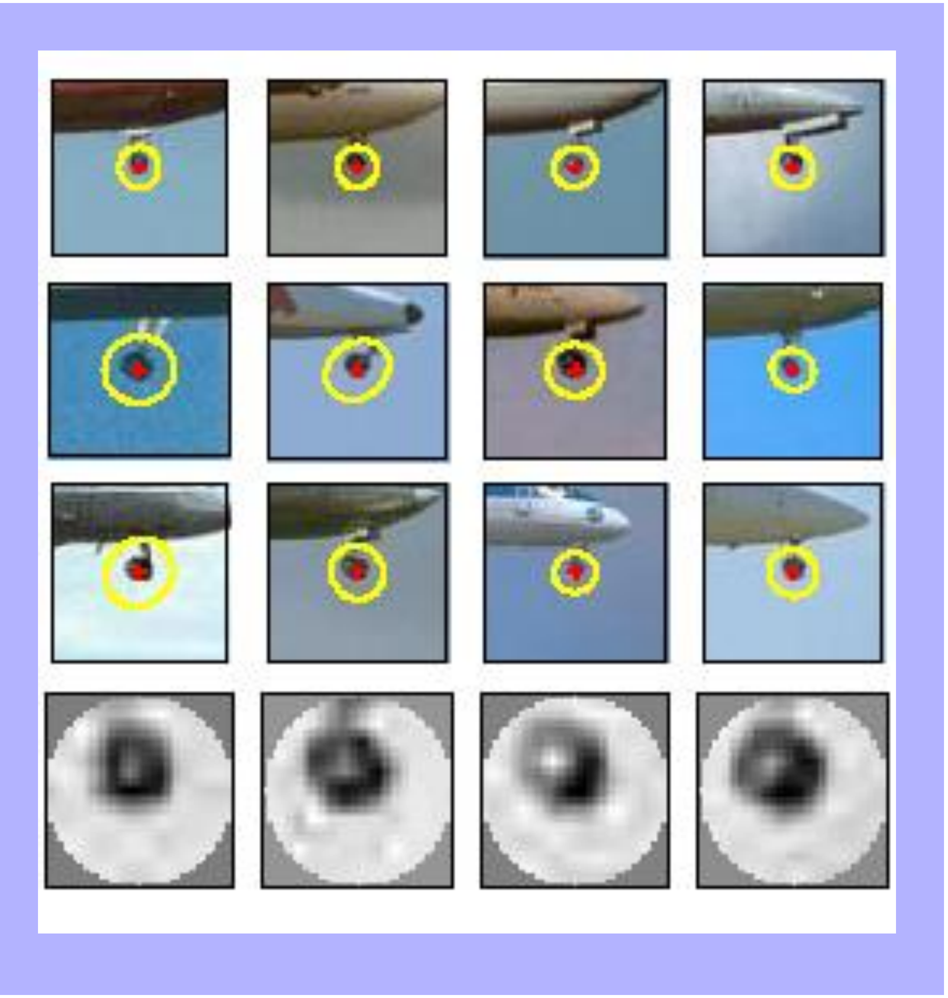
- Almost every aspect of K-means has been modified
  - Distance measures
  - Centroid and objective definitions
  - Overall process
  - Efficiency Enhancements
  - Initialization

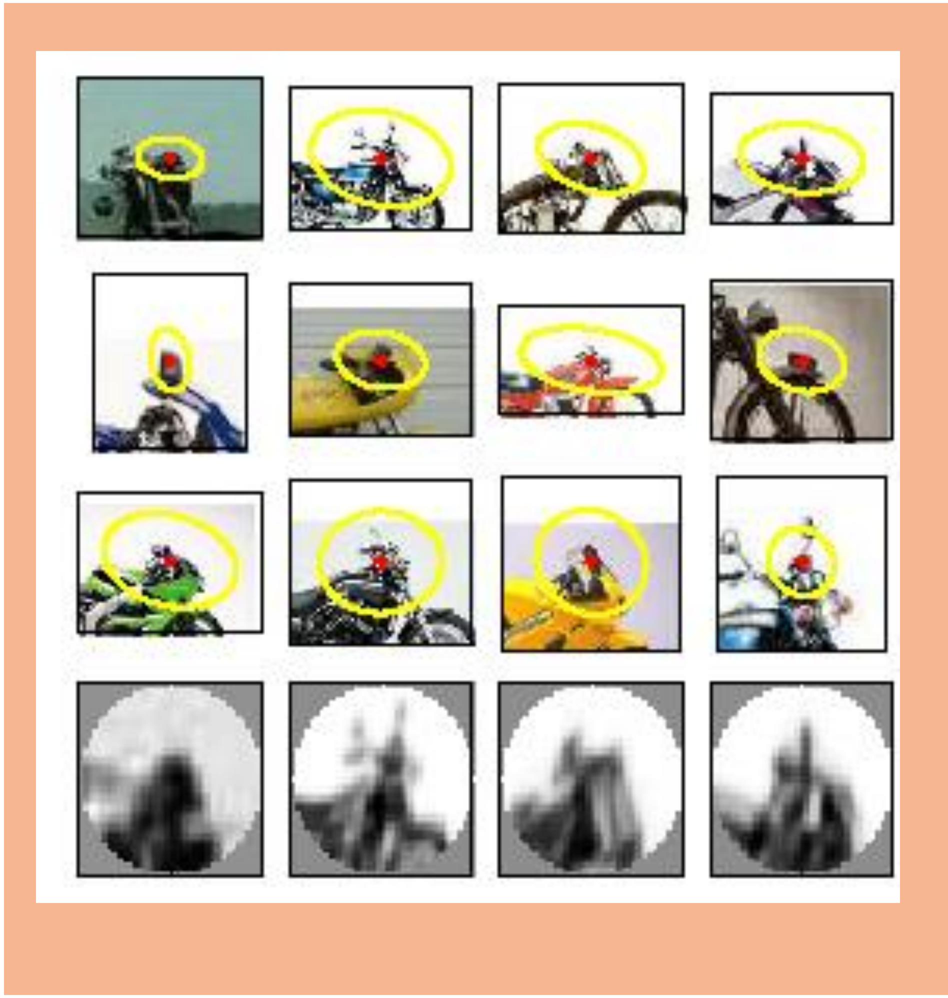
K-means

- Many implementations. (you could try your own)
- We will be using Dollar toolbox http://vision.ucsd.edu/~pdollar/toolbox/
- Kmeans2

```
% USAGE
%  [ IDX, C, d ] = kmeans2( X, k, [varargin] )
%
% INPUTS
%  X       - [n x p] matrix of n p-dim vectors.
%  k       - maximum nuber of clusters (actual number may be smaller)
%  prm     - additional params (struct or name/value pairs)
%   .k         - [] alternate way of specifying k (if not given above)
%   .nTrial    - [1] number random restarts
%   .maxIter   - [100] max number of iterations
%   .display   - [0] Whether or not to display algorithm status
%   .rndSeed   - [] random seed for kmeans; useful for replicability
%   .outFrac   - [0] max frac points that can be treated as outliers
%   .minCl     - [1] min cluster size (smaller clusters get eliminated)
%   .metric    - [] metric for pdist2
%   .C0        - [] initial cluster centers for first trial
%
% OUTPUTS
%  IDX    - [n x 1] cluster membership (see above)
%  C      - [k x p] matrix of centroid locations C(j,:) = mean(X(IDX==j,:))
%  d      - [1 x k] d(j) is sum of distances from X(IDX==j,:) to C(j,:)
%           sum(d) is a typical measure of the quality of a clustering
%
```

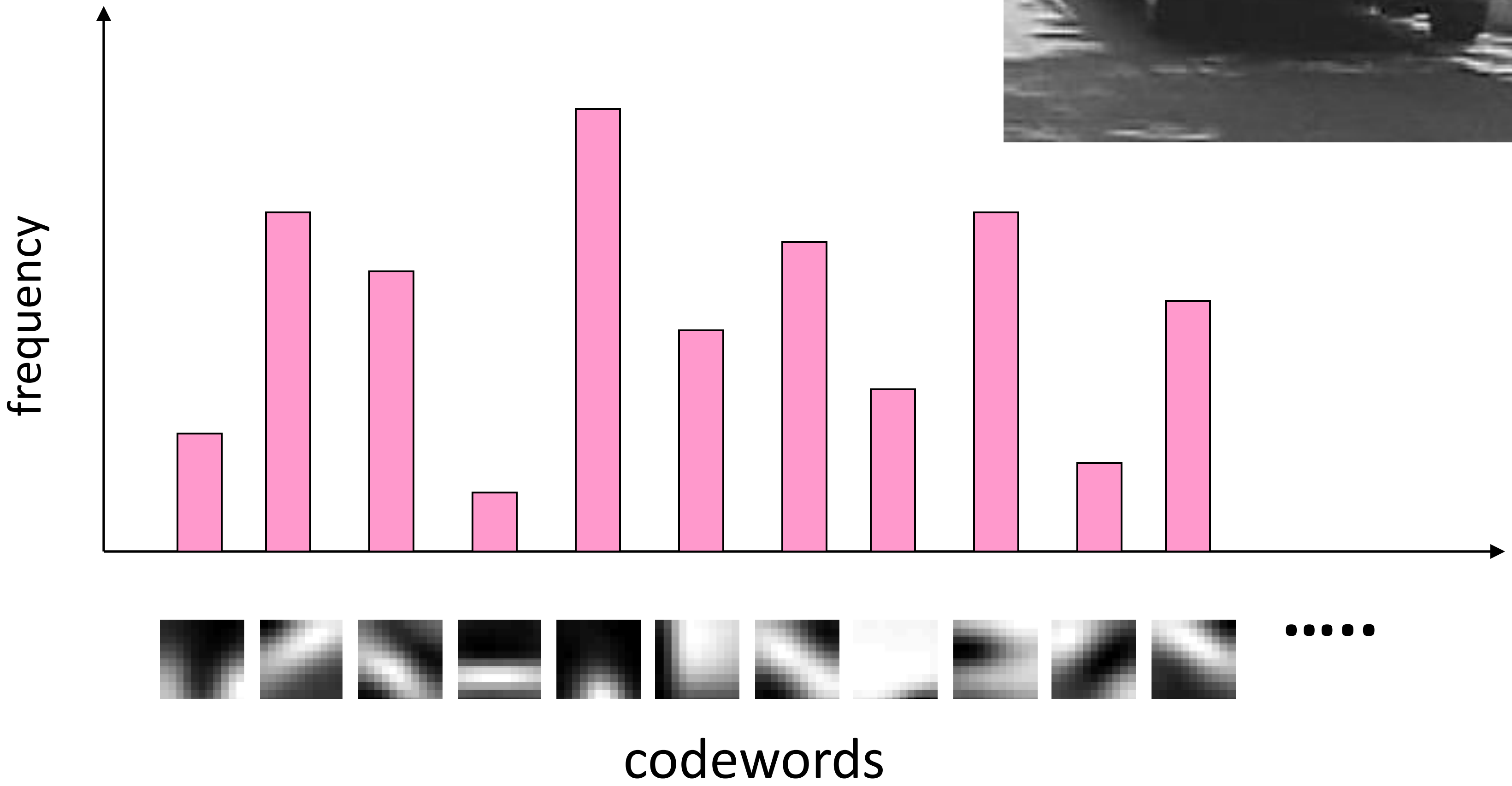
Image patch examples of codewords



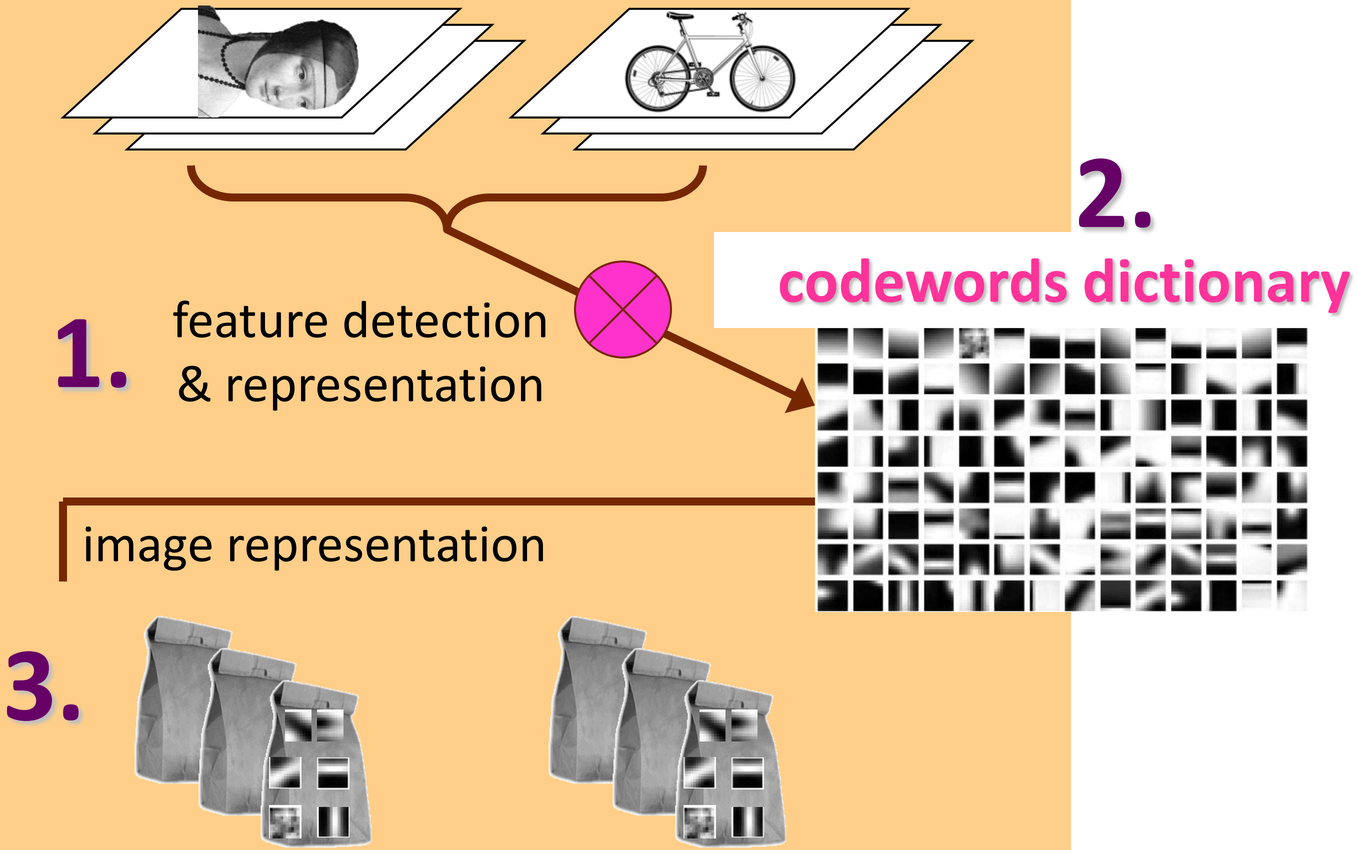


Sivic et al. 2005

3. Image representation



Representation



Learning and Recognition