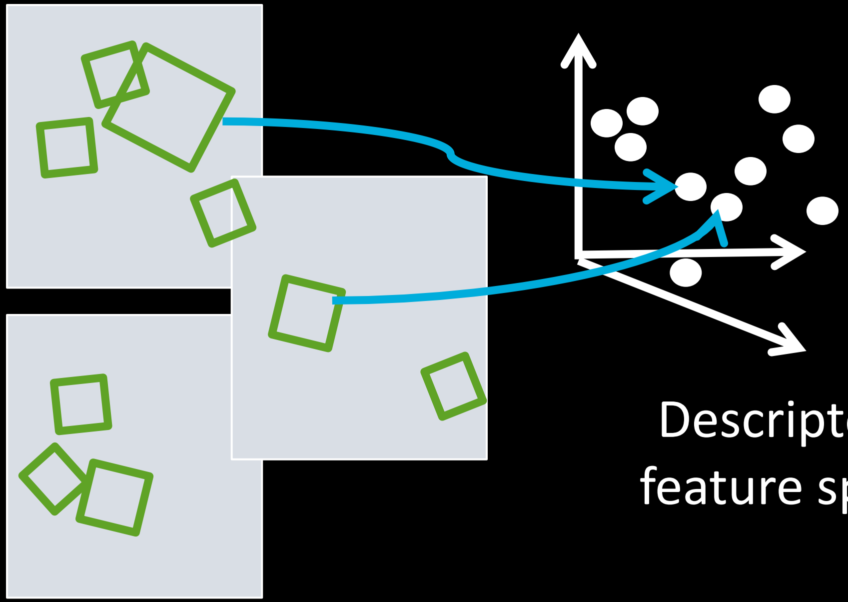
CS4495/6495
Introduction to Computer Vision

8C-L4 *Bag of visual words*

Indexing local features

- Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)

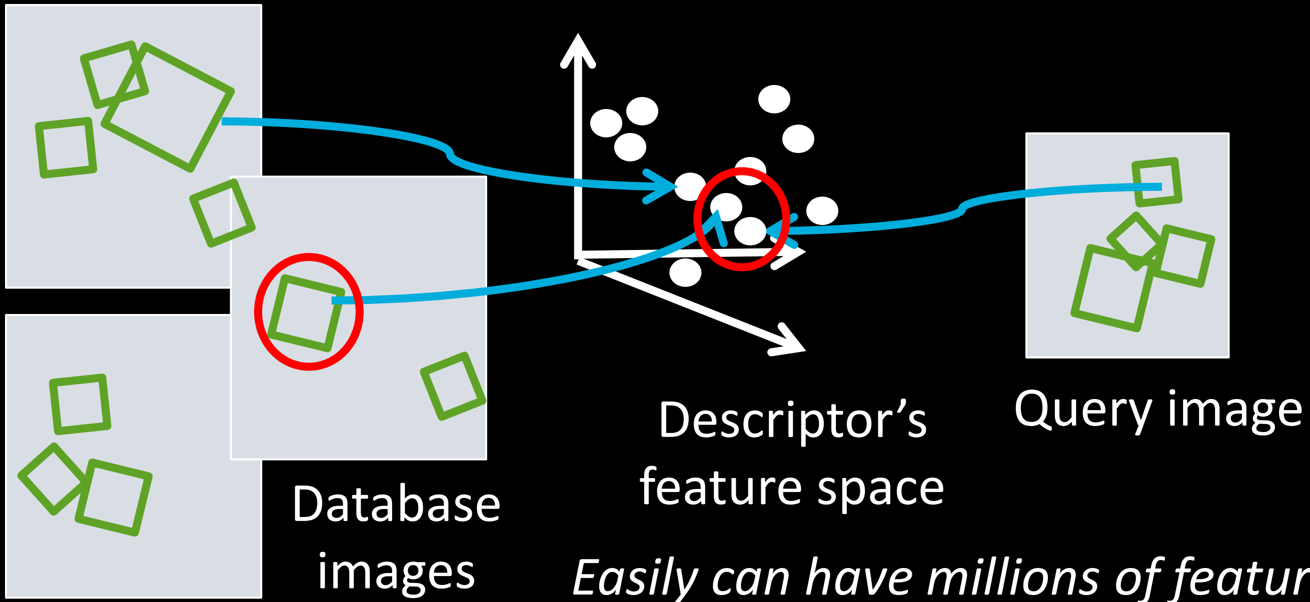


Descriptor's feature space

Kristen Grauman

Indexing local features

- When we see close points in feature space, we have similar descriptors, which indicates similar local content.



Easily can have millions of features to search!

Indexing local features

- With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?

Kristen Grauman

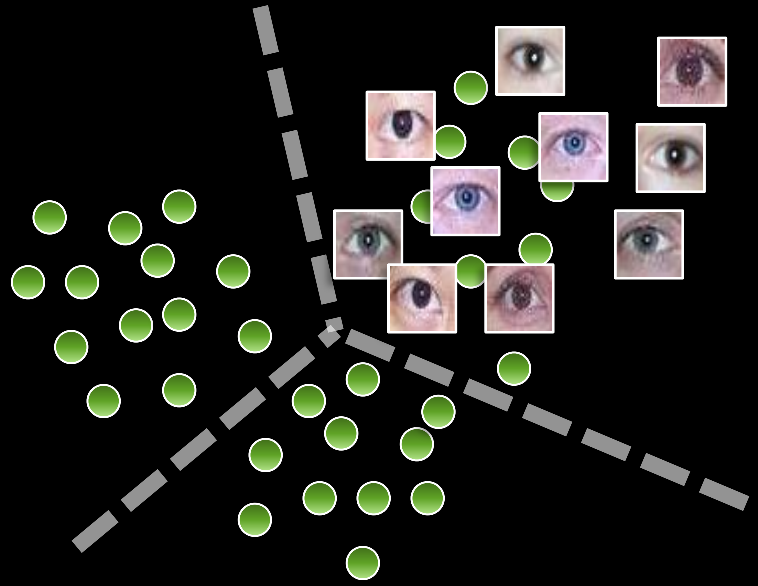
Indexing local features: inverted file index

- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index…
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we'll need to map our features to "visual words"

Kristen Grauman

Visual words (discretization)

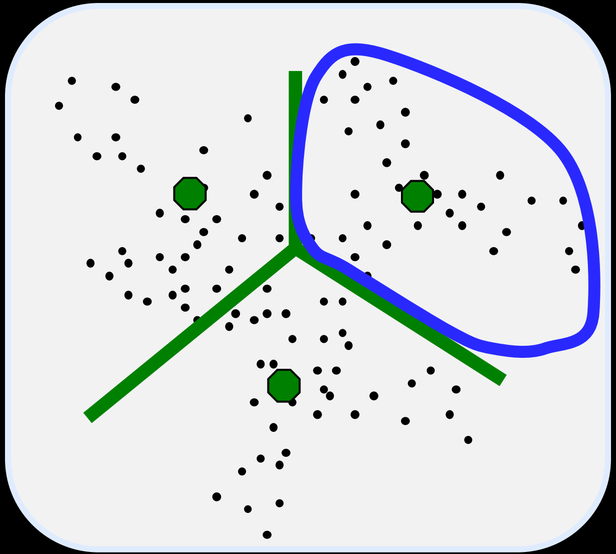
- Words are discrete, visual features are typically continuous…



Discretization via clustering/vector quantization

Visual words

Example: each group of patches belongs to the same visual word



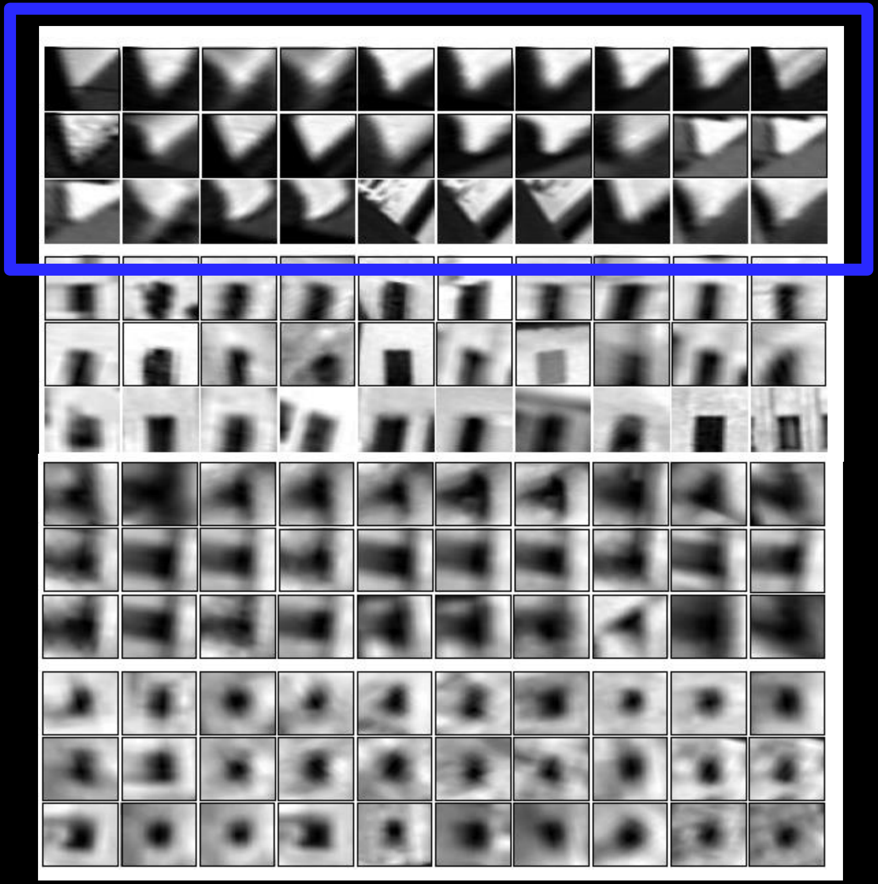
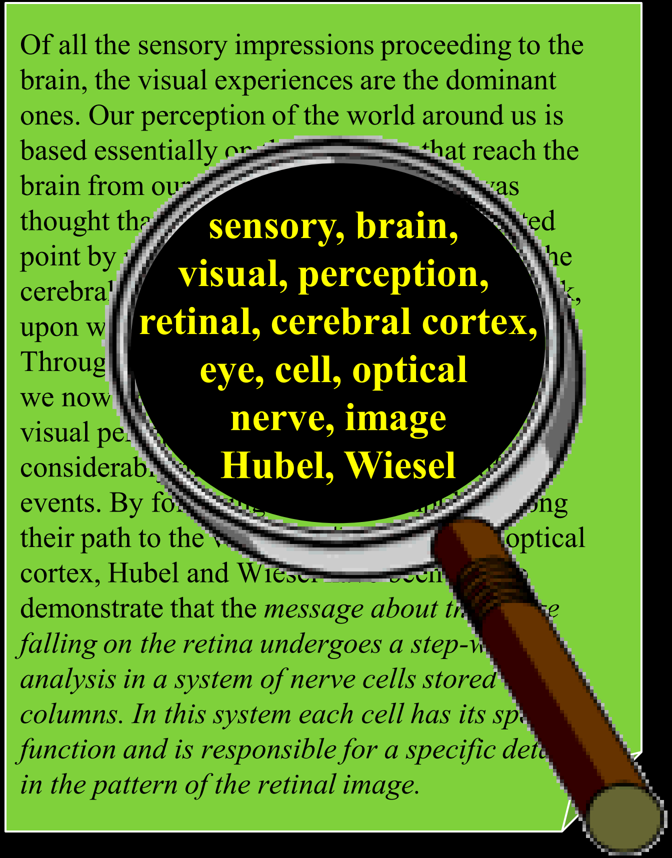
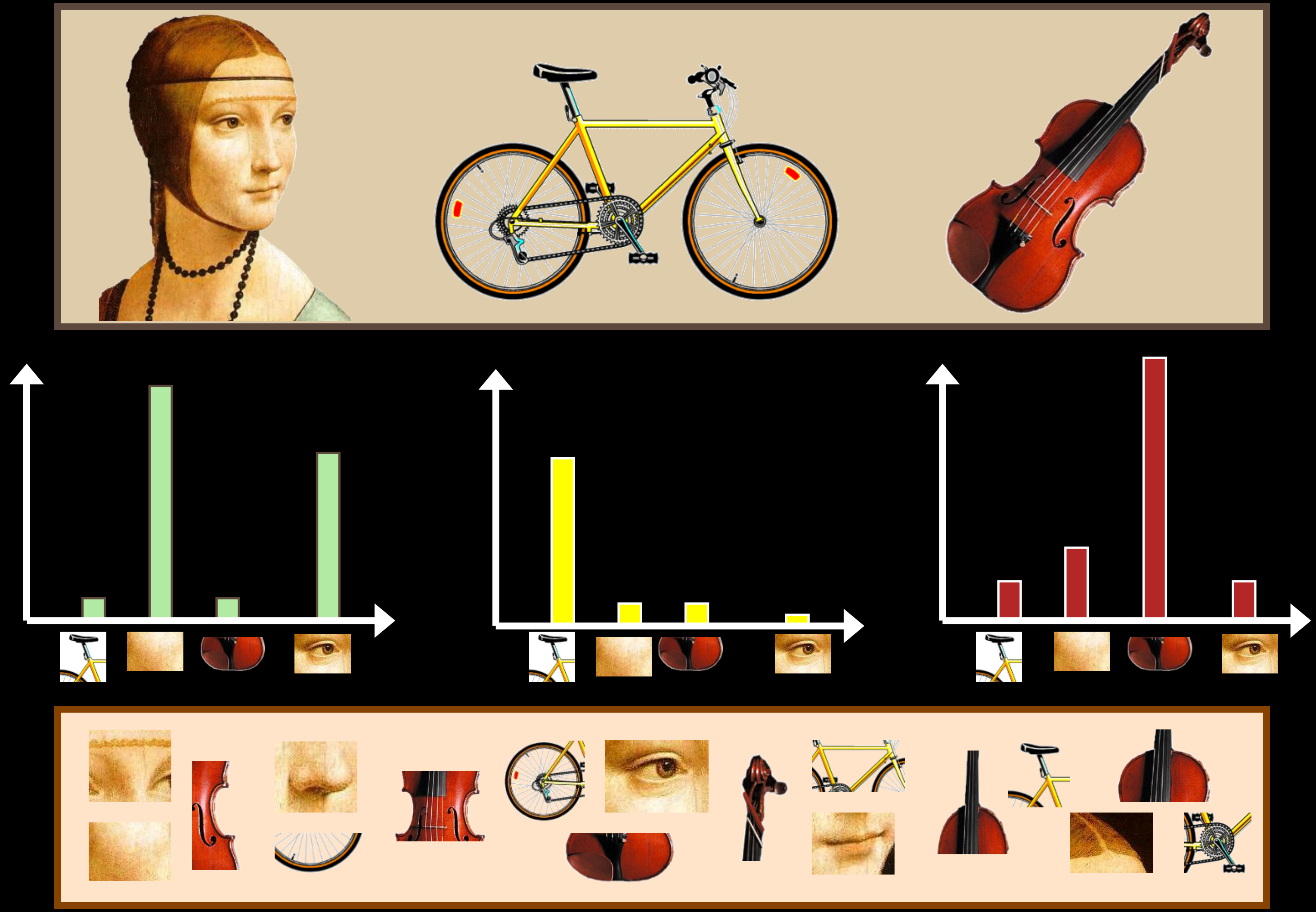


Figure from Sivic & Zisserman, ICCV 2003

Analogy to documents

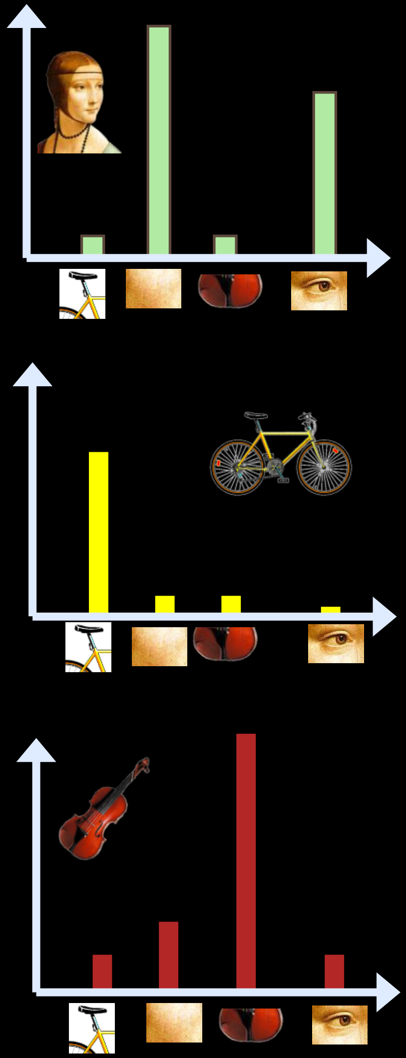


Object → **Bag of ‘words’**



Bags of visual words

- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.

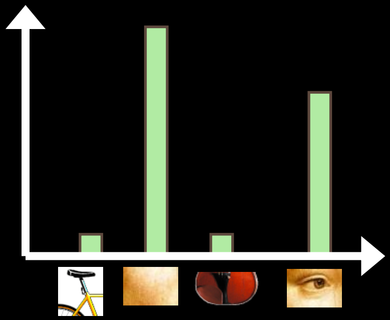


Comparing bags of words

- Rank by normalized scalar product between their (possibly weighted) occurrence counts-- nearest neighbor search for similar images.

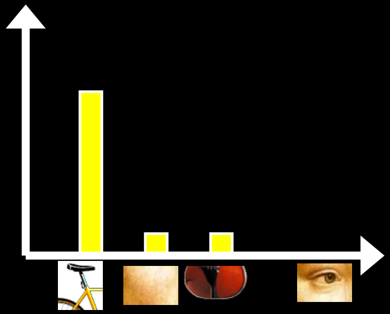
Comparing bags of words

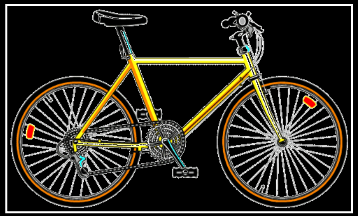
[1 8 1 4]



d_j

[5 1 1 0]





q

$$sim(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

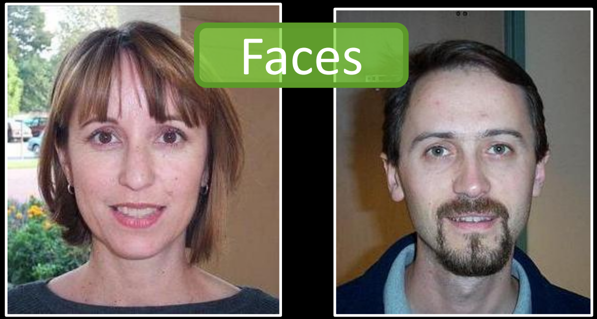
$$= \frac{\sum_{i=1}^{V} d_j(i) * q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} * \sqrt{\sum_{i=1}^{V} q(i)^2}}$$

for vocabulary of V words

Kristen Grauman

Object classification with bag of words

- Performance on Caltech 101 dataset with linear SVM on bag-of-word vectors:



True classes →	*faces (frontal)*	*airplanes (side)*	*cars (rear)*	*cars (side)*	*motorbikes (side)*
faces(frontal)	**94**	0.4	0.7	0	1.4
airplanes (side)	1.5	**96.3**	0.2	0.1	2.7
cars (rear)	1.9	0.5	**97.7**	0	0.9
cars(side)	1.7	1.9	0.5	**99.6**	2.3
motorbikes (side)	0.9	0.9	0.9	0.3	**92.7**
Mean ranks	1.07	1.04	1.03	1.01	1.09

[Csurka et al., ‘04]