

From Shallow to Deep representation for multimedia data classification

Lecture # 3
*BoW and extensions: coding, pooling & spatial
pyramid*
2017-2018
Frederic Precioso

Analogy to documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially upon what we see. Light reaches the brain from our eyes, and it is here that the visual system begins. We have already seen that the visual system is complex, and that it is not until the optic nerve reaches the brain that the visual information is fully processed. In 1960, Hubel and Wiesel performed experiments on the visual system of the cat, and they found that the visual information was processed in a series of layers in the cerebral cortex. They were able to demonstrate that the visual system worked in a step-wise fashion, with each layer receiving input from the previous one. This analogy can be extended to the visual system of the human brain. The visual system is composed of several layers, each with a specific function. The first layer is the retina, which receives light from the eye and converts it into electrical signals. These signals are then transmitted to the brain via the optic nerve. The brain then processes these signals to produce a visual perception. This analogy can be extended to the visual system of the human brain. The visual system is composed of several layers, each with a specific function. The first layer is the retina, which receives light from the eye and converts it into electrical signals. These signals are then transmitted to the brain via the optic nerve. The brain then processes these signals to produce a visual perception.

**sensory, brain,
visual, perception,
retinal, cerebral cortex,
eye, cell, optical
nerve, image
Hubel, Wiesel**

China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be driven by a 30% jump in exports to \$150bn and a 30% rise in imports to \$60bn. The ministry said further analysis showed China's trade surplus factor. It said the country's domestic demand was the main driver of the country. China has been buying against the dollar to support its exports, and it has committed itself to trade within a narrow range. The ministry wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take time and tread carefully before allowing the yuan to rise further in value.

**China, trade,
surplus, commerce,
exports, imports, US,
yuan, bank, domestic,
foreign, increase,
trade, value**

Bag of Word: introduction

- Requires to find the same object with deformations
- Winning combinaison: sparse PoI detector + discriminant and invariant descriptors like SIFT
- Local => intrinsically robust to occlusions and noise (i.e. the background)
- Challenge: fast search of nearest neighbors in the very large scale datasets
- Geometrical checking in order to constraint the matches

Bag of Word model (BoW) from text retrieval

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based on the messages that reach the brain from the eyes. For a long time it was believed that the image was transmitted directly to the brain. It turns out that the image is processed in the brain before it reaches the visual cortex. The visual system consists of three main parts: the eye, the optic nerve, and the brain. The eye receives light from the environment and converts it into electrical signals. These signals are sent along the optic nerve to the brain. In the brain, the signals are processed by various regions, including the visual cortex. The visual cortex is responsible for interpreting the signals and generating the perception of the visual world.

sensory, brain, visual, perception, retinal, cerebral cortex, eye, cell, optical nerve, image Hubel, Wiesel

China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be fuelled by a predicted 30% jump in exports, which would be offset by a 18% rise in imports. Analysts say the figures are likely to further fuel concerns that Beijing is manipulating the yuan. Last week, the US Treasury argued that China's central bank, the People's Bank, deliberately undervalues the yuan. The US says the surplus is only justified because China's economy is still catching up with the rest of the world. The US wants the yuan to appreciate so that it can do more to encourage exports. The US has threatened to impose trade sanctions if the yuan does not appreciate. The Chinese government has responded by saying that it will not allow the yuan to appreciate too quickly. It has also said that it will not allow the yuan to depreciate either. The Chinese government has said that it will take its time and think carefully before allowing the yuan to rise further in value.

China, trade, surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value

We compare two documents by comparing their histograms of occurrence of words

Documents (Google)

Create search engines understanding natural language

-> evolution of *Google* beyond keywords

-> « text-mining » or « web-mining »

Collection: ensemble of documents

Vector Space Model (VSM): N -dimension space with N , the number of useful terms from the language (“the”, “of”...are useless and are thus called **stop-words**). The space is so split into N clusters...

Documents (cont'd)

- Step 1: For each of the N terms of the language, create an **Inverse Index** which store, for each term of the language the documents containing this term
 - Computation of **Document Frequency** = nb doc using this term / total document = **df** (greater the df, the less the term has importance from an informative point of view)
 - soit **idf** = $\log(1/\text{df})$
- Step 2: for each document,
 - for each term, computation of the **Term Frequency** (greater tf is in a document, more this term must be important to the subject of the document)
 - Computation of a feature vector **VSM** :
 - For i from 0 to N , $VSM[i] = \text{tf}(\text{term } i \text{ in this document}) * \text{idf}(\text{term } i)$
 - We normalize this vector to get $\|VSM(\text{document})\| = 1$

Documents (cont'd)

- If you want to be more statistician = measures of proximity between 2 vectors-distributions
- For the documents (a feature = a word)...

Table 2: Tableau de contingence

		Mots						
		m_1	...	m_j	...	m_L	somme	
		D_1	$f_{1,1}$...	f_{1j}	...	f_{1L}	...
	
Textes	D_i	f_{i1}	...	f_{ij}	...	f_{iL}	$f_{i\cdot} = \sum_{j=1}^L f_{ij}$	

	D_D	f_{D1}	...	f_{Dj}	...	f_{DL}	...	
	somme	$f \cdot j = \sum_{i=1}^D f_{ij}$	f	

Documents (cont'd)

- In the context of text mining, the **distance du χ^2** ($chis$ or $khis$) is the Euclidean distance between two vectors-documents normalized by their length (number of words),

$$d_{eNid}^2(VSM_1, VSM_2) = \sum_{j=1}^L \left(\frac{f_{1j}}{f_{1.}} - \frac{f_{2j}}{f_{2.}} \right)^2$$

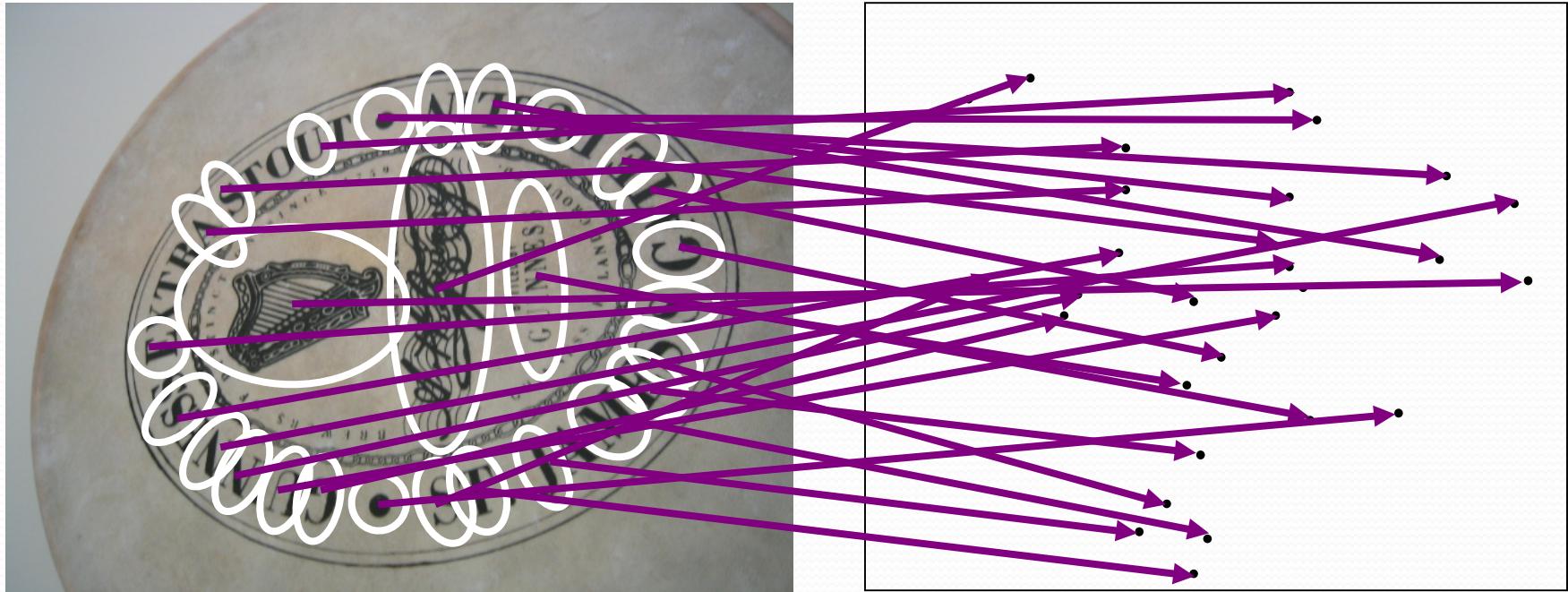
weighted by the mass of each of the words compared to all of the texts
(total number of appearance of a word in all of the texts)

$$\chi^2(VSM_1, VSM_2) = \sum_{j=1}^L \frac{f}{f_{.j}} \left(\frac{f_{1j}}{f_{1.}} - \frac{f_{2j}}{f_{2.}} \right)^2$$

- f corresponds to the total number of words in the set of documents
- f_{ij} corresponds to the frequency of the word j in document i
- $f_{i.}$ corresponds to the total number of words in document i
- $f_{.j}$ corresponds to the frequency of the word j in the set of documents
- Very useful whenever you want to compare histograms, probability distributions...

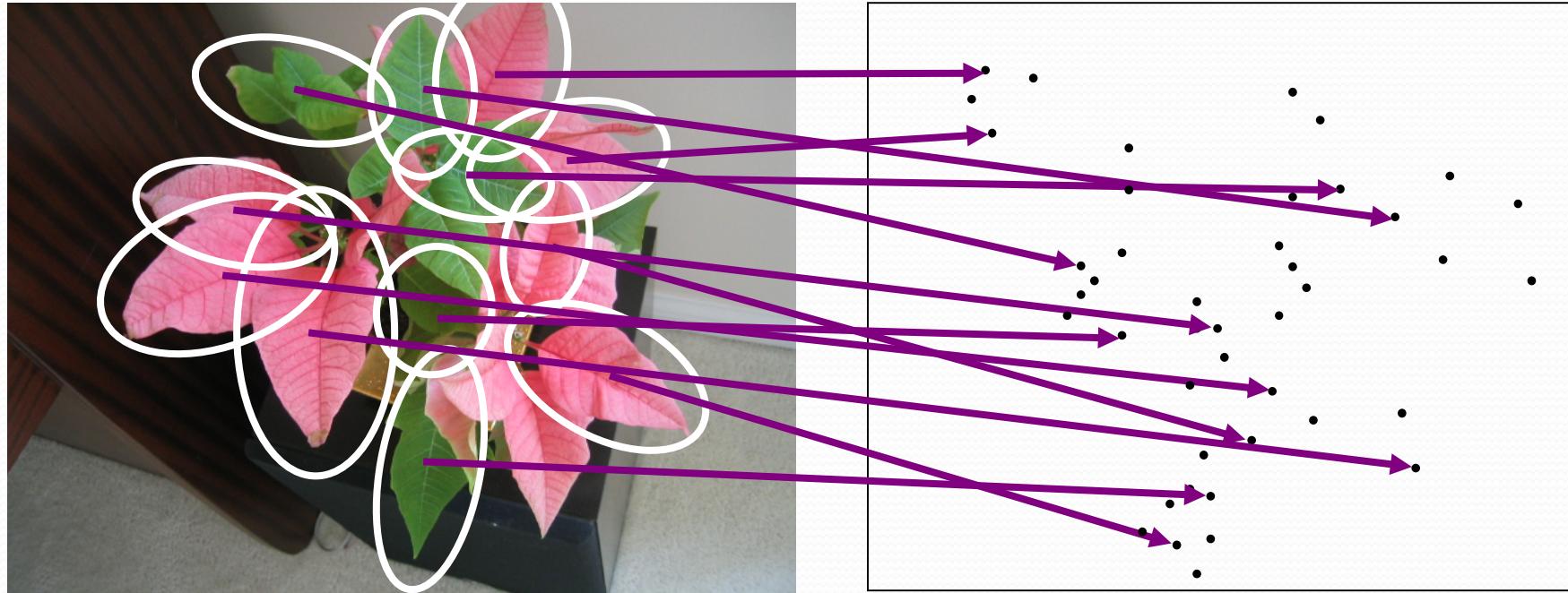
Visual words: main idea

- Extract some local features from a number of images ...

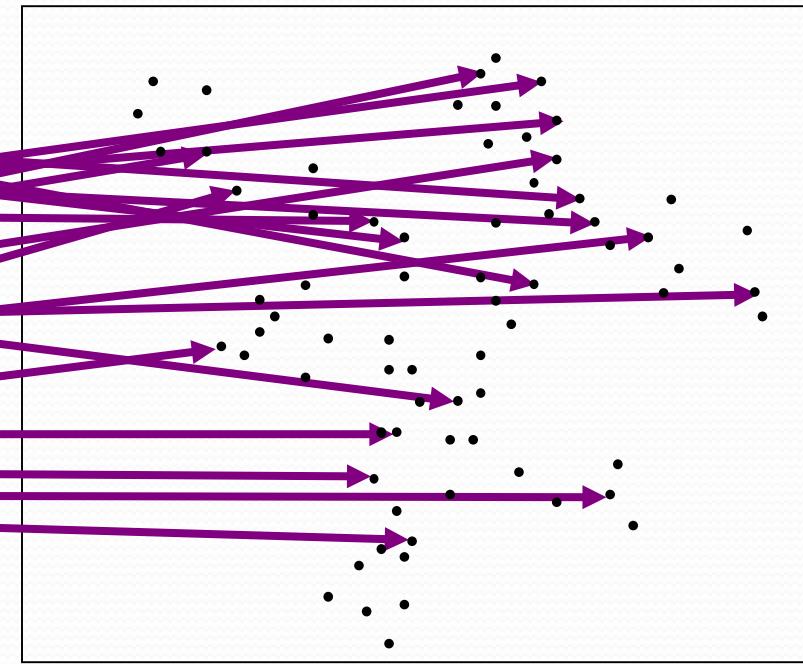


e.g., SIFT descriptor space:
each point is 128-
dimensional

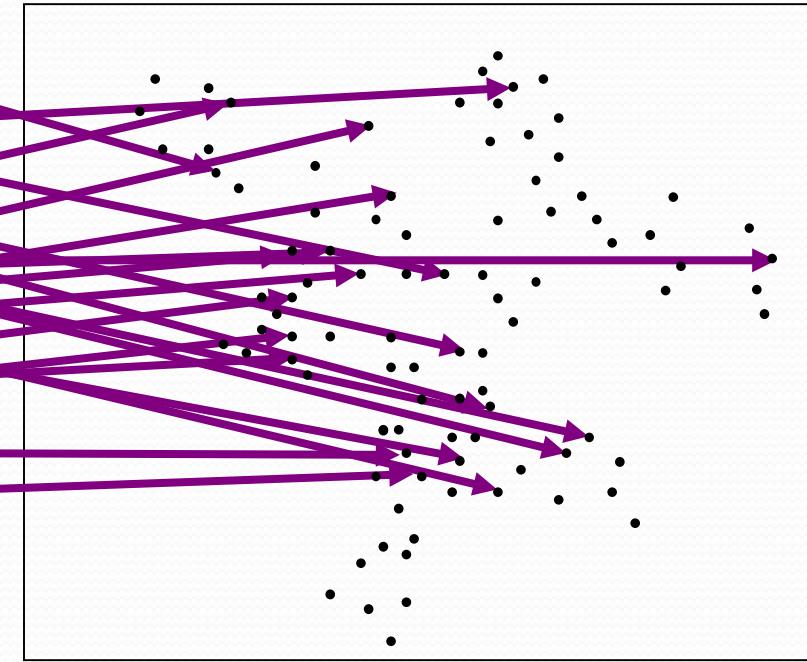
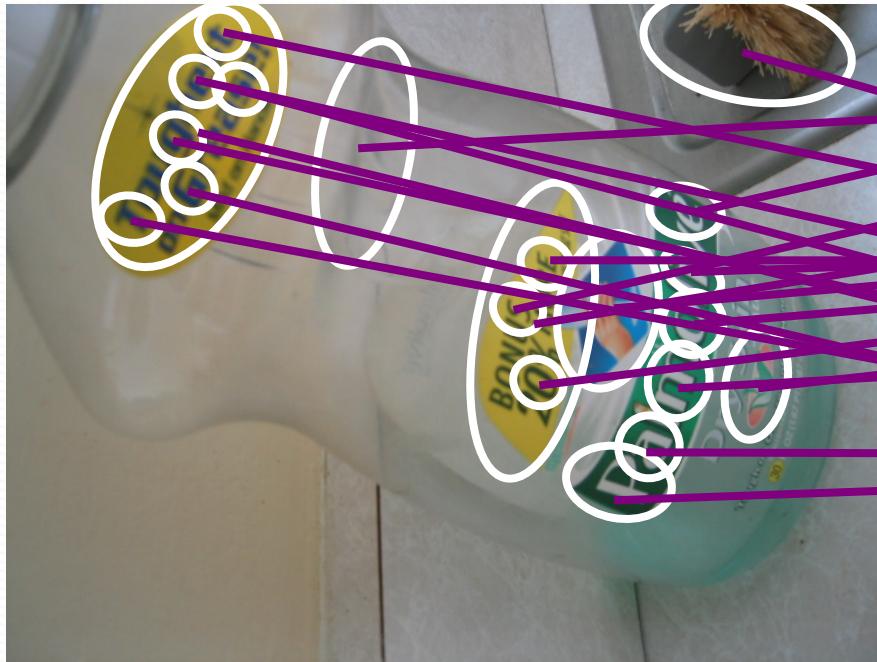
Visual words: main idea

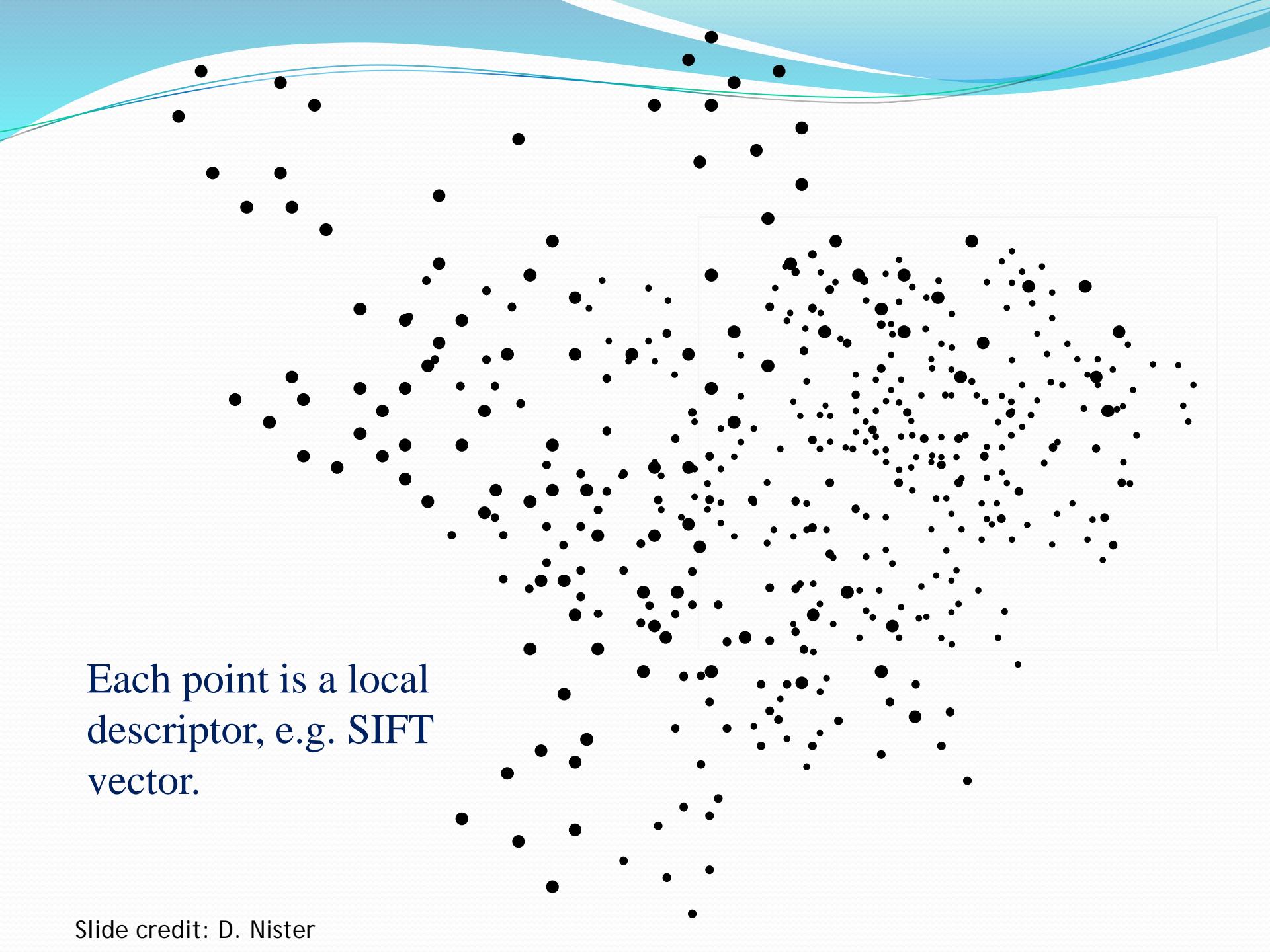


Visual words: main idea

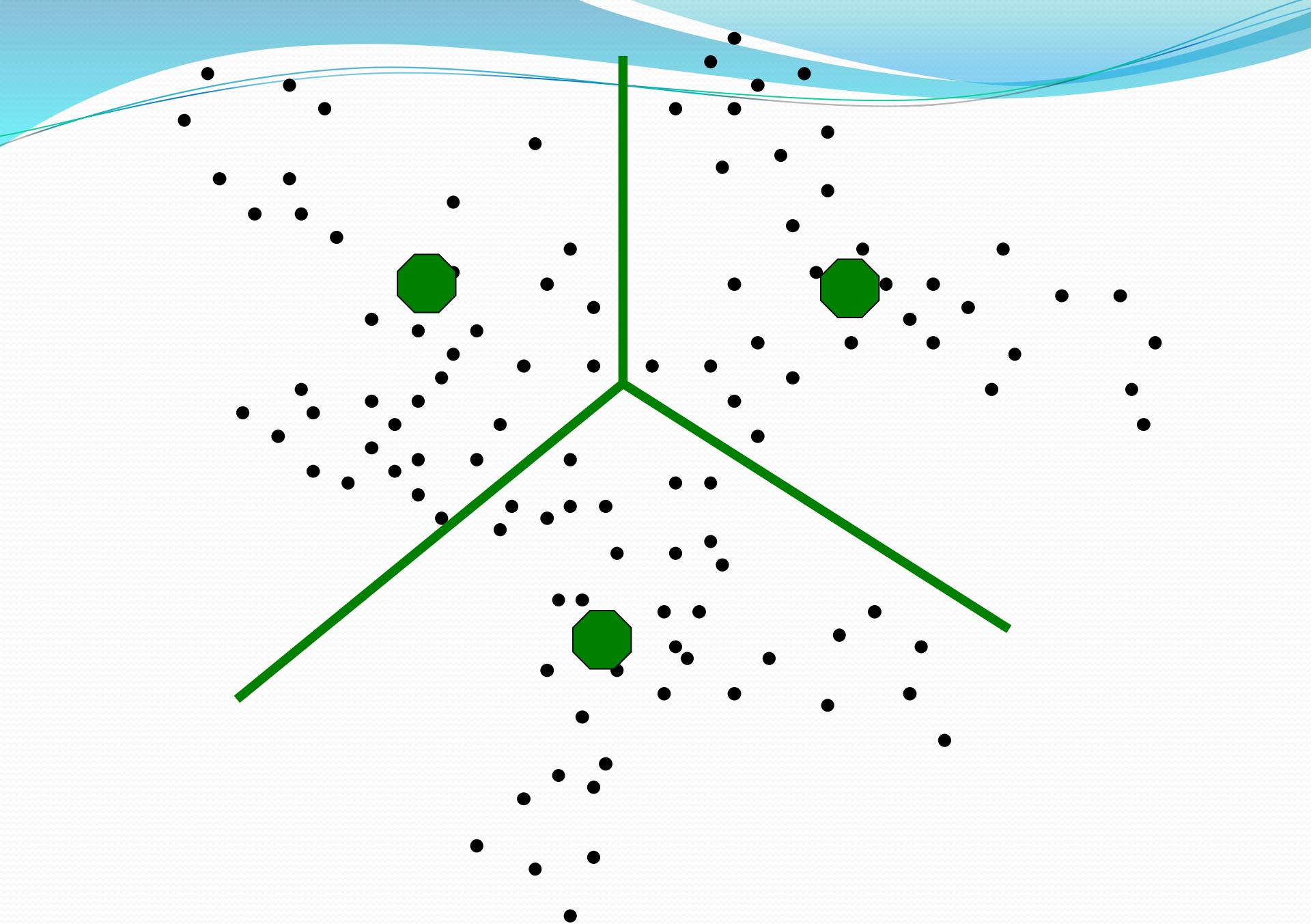


Visual words: main idea





Each point is a local descriptor, e.g. SIFT vector.



Visual words

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

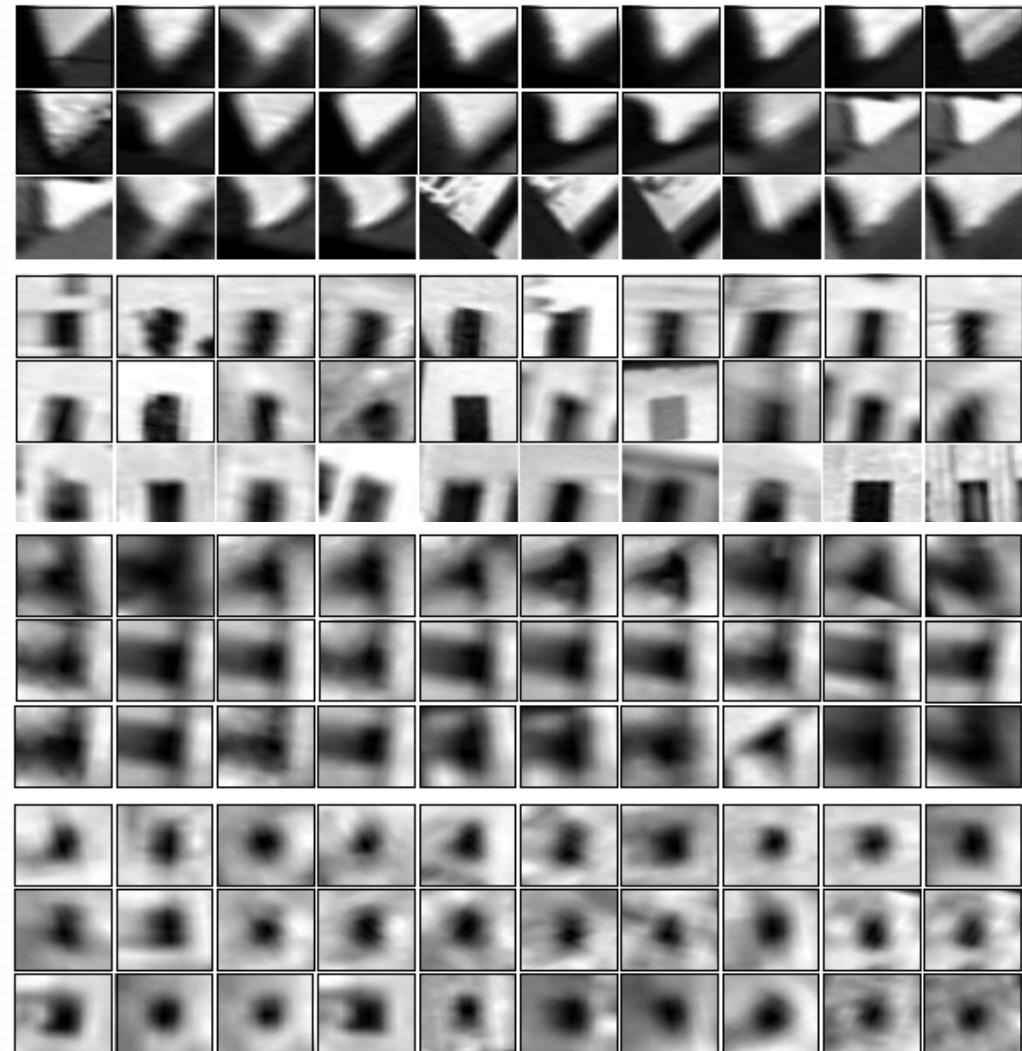
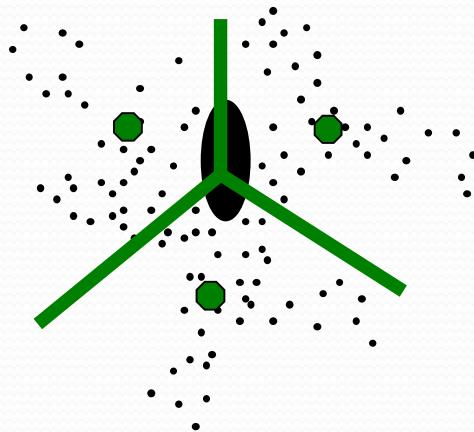
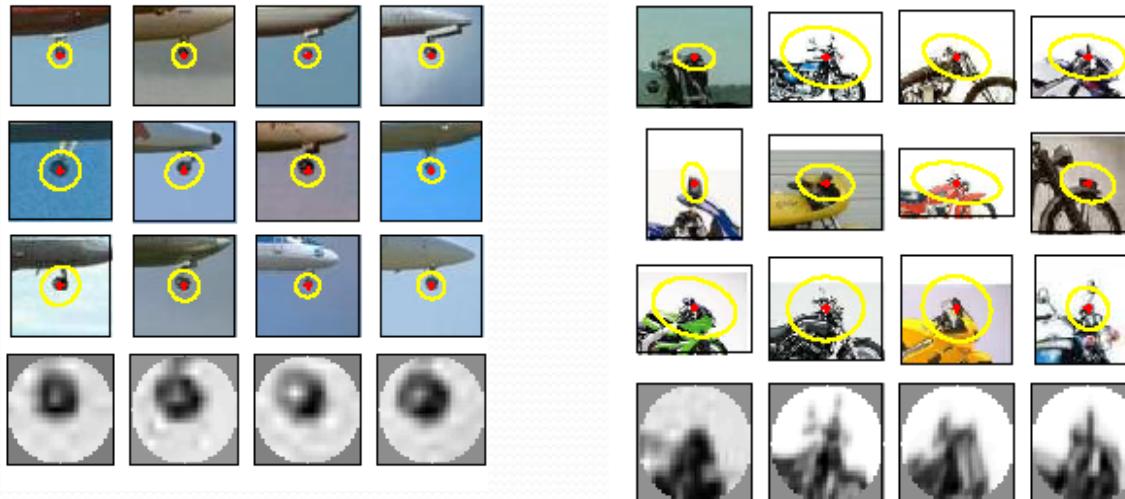


Figure from Sivic & Zisserman, ICCV 2003

Visual words

- They have also been used for describing scenes and objects for the sake of indexing or classification.

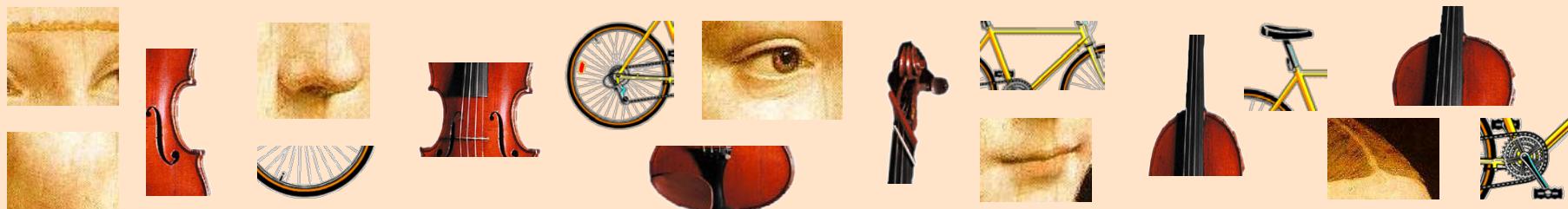
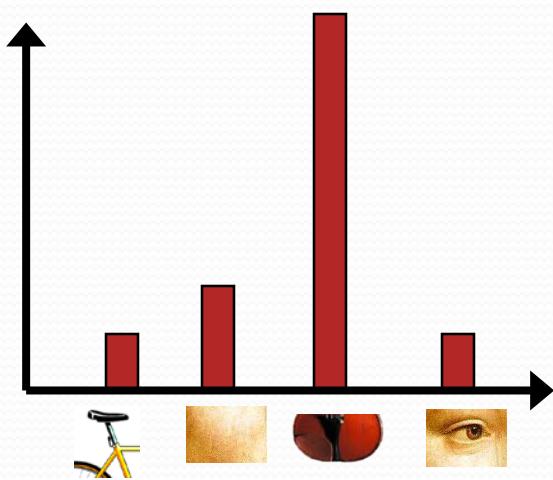
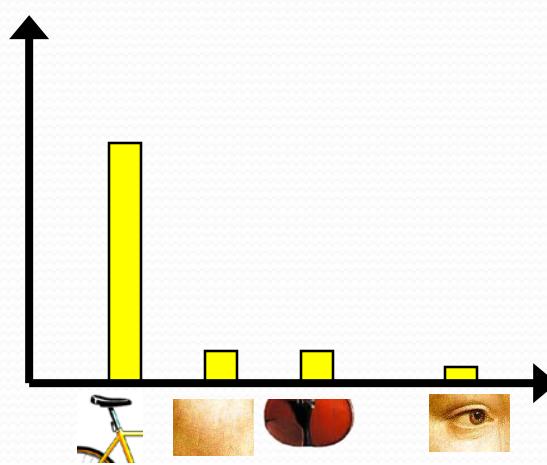
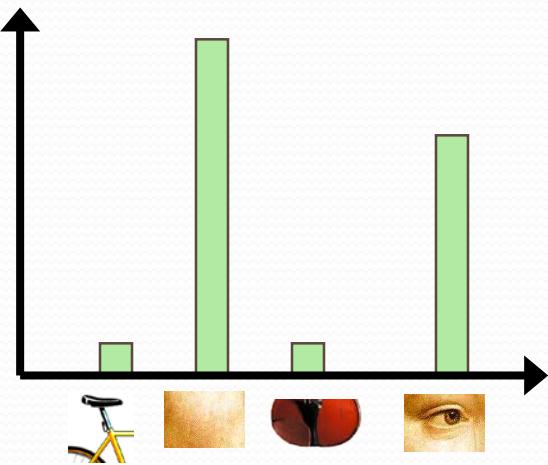


Sivic & Zisserman 2003; Csurka,
Bray, Dance, & Fan 2004; many
others.

Object

Bag of ‘words’

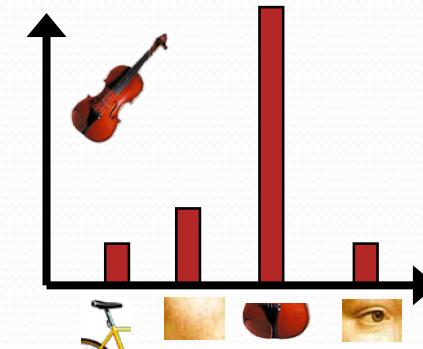
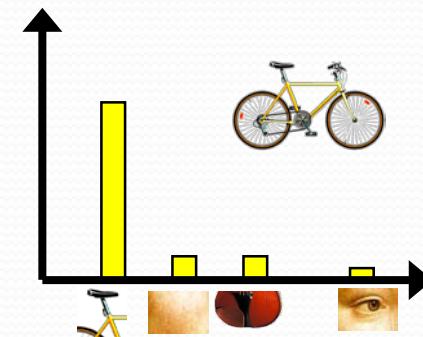
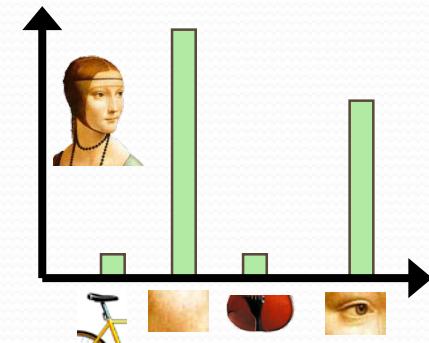




Bags of visual words



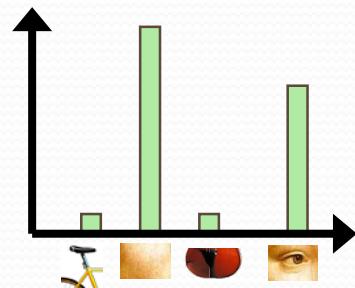
- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.



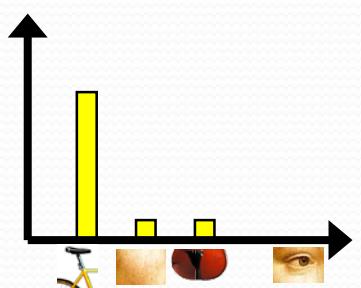
Comparing bags of words

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

[1 8 1 4]



[5 1 1 0]



$$\begin{aligned} sim(d_j, q) &= \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|} \\ &= \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \times \sqrt{\sum_{j=1}^t w_{i,q}^2}} \end{aligned}$$



\vec{d}_j

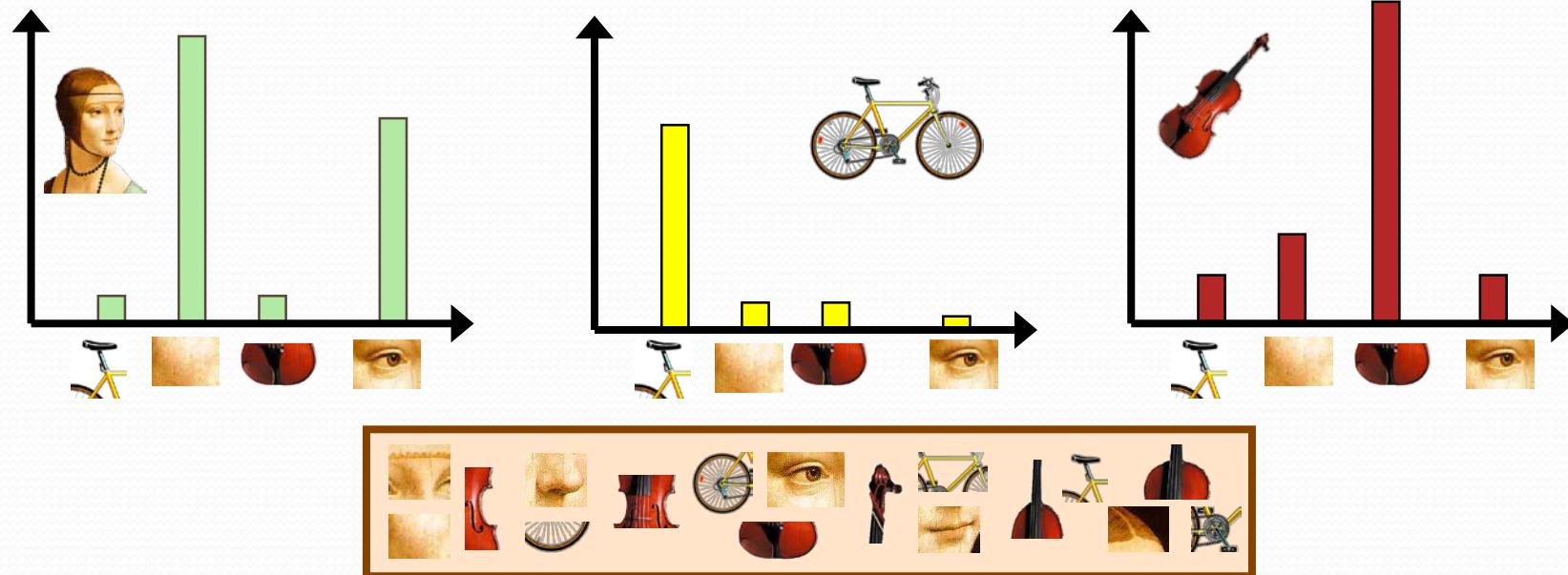
\vec{q}

Bag of Visual Words: online part

2 stages:

- Coding: projection of each local descriptor => dictionary
- Pooling (projection aggregation): statistical summary / Visual Word

=> Image global index: likelihood to contain each Visual Word



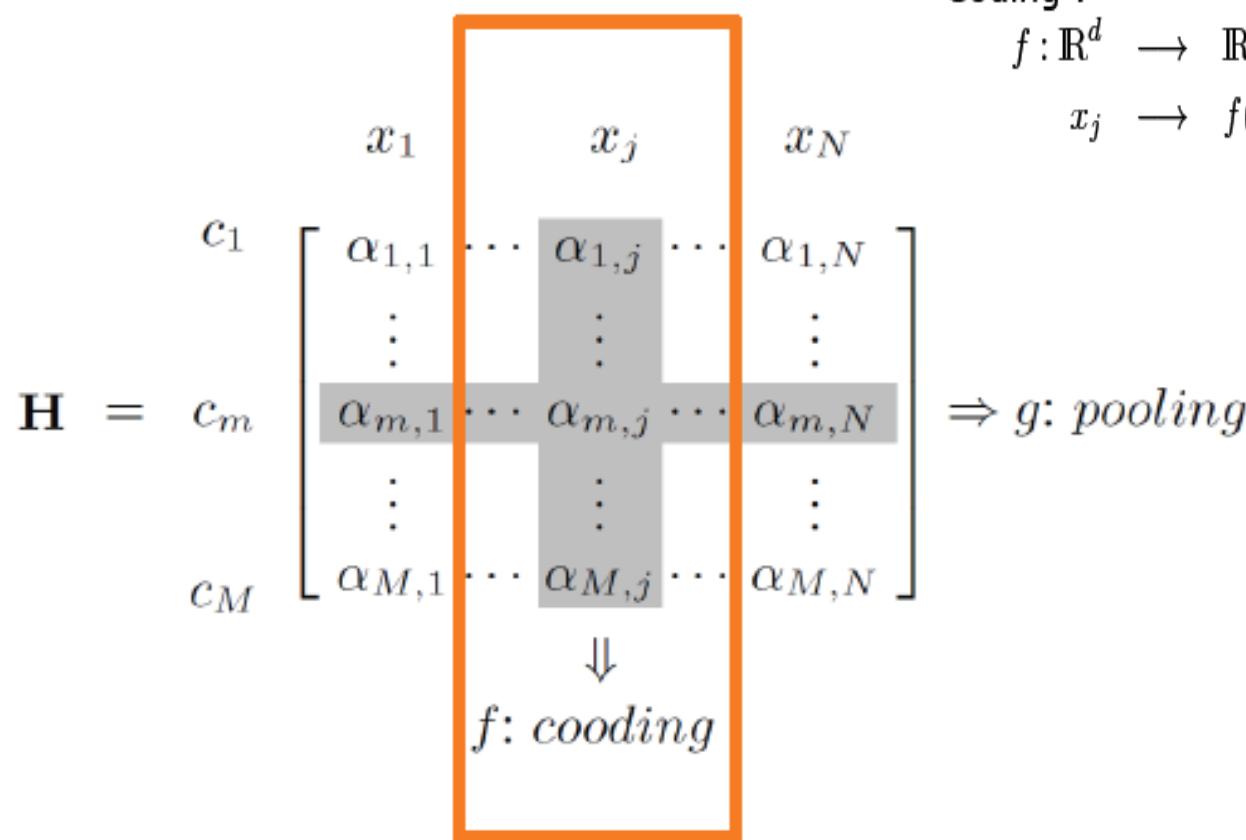
Coding

Notations :

- Les données images : $X = \{x_j \in \mathbb{R}^d\}, j \in \{1; N\}$
- Les centres : $C = \{C_m\}, m \in \{1; M\}$
- Coding :

$$f: \mathbb{R}^d \rightarrow \mathbb{R}^M$$

$$x_j \rightarrow f(x_j) = \alpha_j = \{\alpha_{m,j}\}, \quad m \in \{1; M\}$$



Coding

- Hard assignment: the simplest (cf text retrieval)

Hard coding: $f = f_Q$ assigns a constant weight to its closest center:

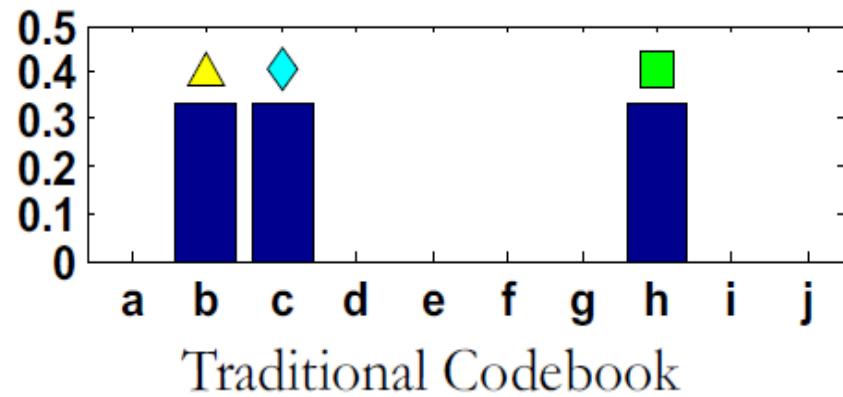
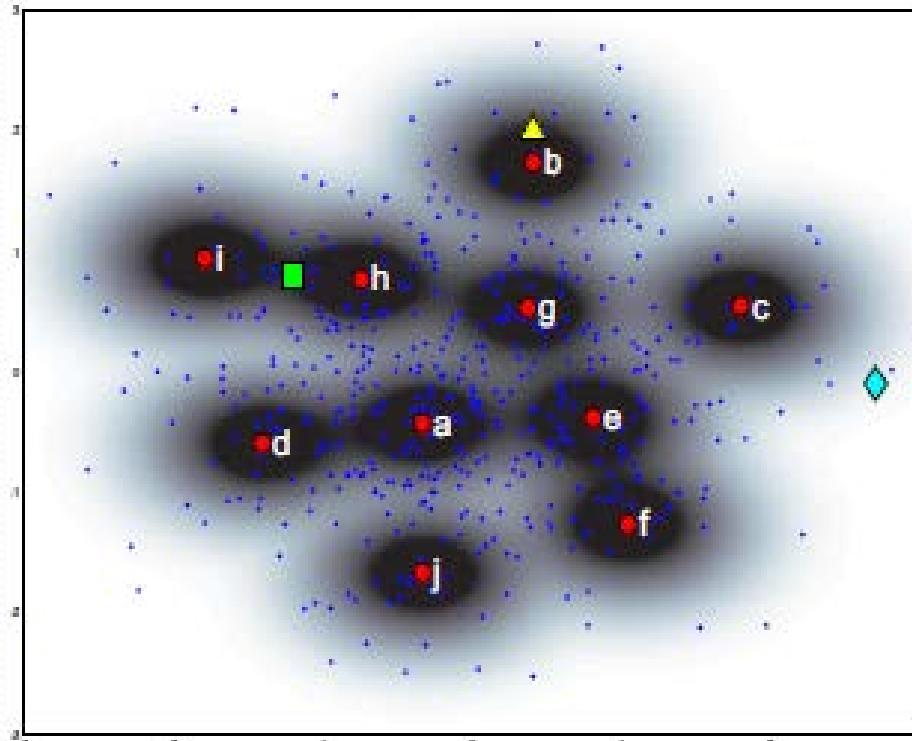
$$f_Q(x_j)[m] = \begin{cases} 1 & \text{if } m = \underset{k \in \{1;M\}}{\operatorname{argmin}} \|x_j - c_k\|^2 \\ 0 & \text{otherwise} \end{cases}$$

- To be more precise
 - Soft assignment
 - Sparse coding
 - Importance of the locality in coding

=> Image global index: likelihood to contain each Visual Word

Projection local signature => dictionary

- The idea the simplest: hard assignment
 - We seek the nearest cluster of the descriptor
 - We assign to this cluster a fixed weight (e.g. 1)



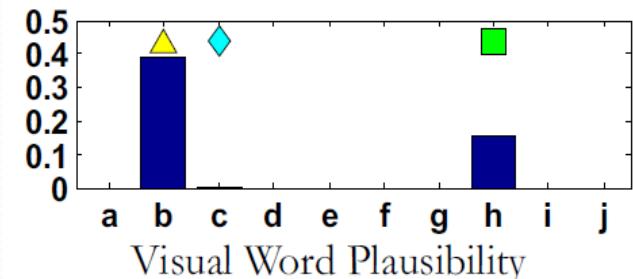
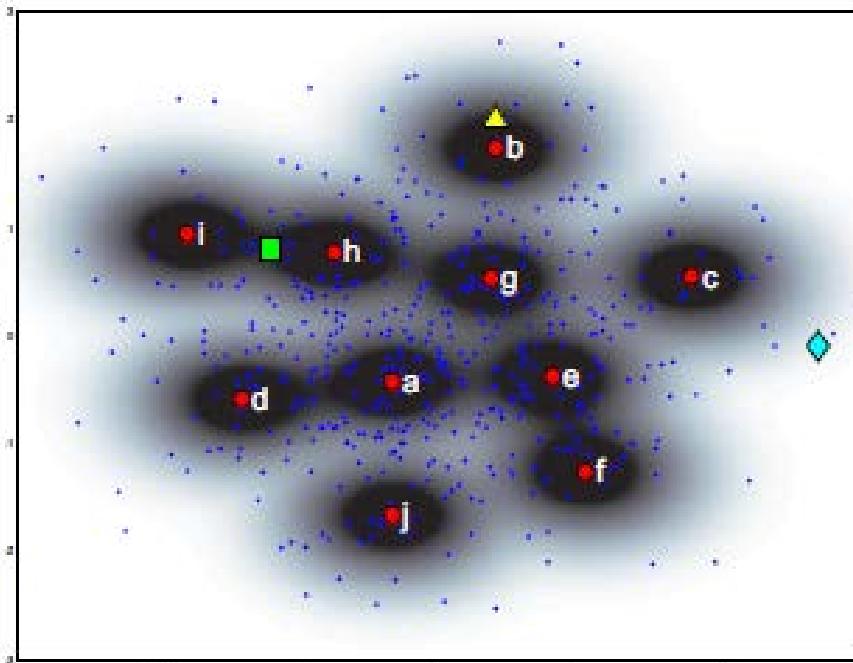
Projection local signature => dictionary

- More sophisticated: soft assignment

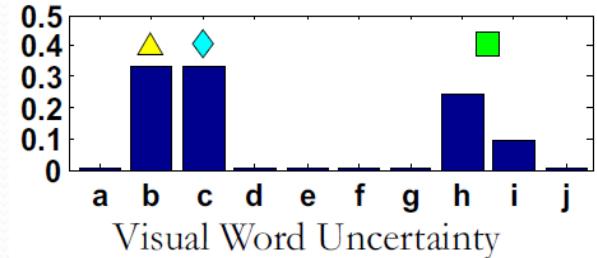
	Best Candidate	Multiple Candidates
Constant Weight	Traditional Codebook	Codeword Uncertainty
Kernel Weighted	Codeword Plausibility	Kernel Codebook

TABLE I

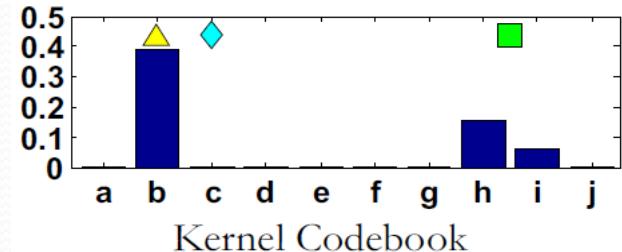
THE RELATIONSHIP BETWEEN VARIOUS FORMS OF CODEWORD AMBIGUITY AND THEIR PROPERTIES.



Visual Word Plausibility signifies that an image feature may not be close enough to warrant representation by any relevant codeword in the vocabulary



Visual Word Uncertainty indicates that one image region may distribute probability mass to more than one codeword.



Projection local signature => dictionary

- **Soft vs hard assignement**
 - In practice, the gain of the soft / hard is not always clear
 - Only the strategy based on *Uncertainty* improves the performance of classification
- **Other approach: sparse coding**
 - We approximate each local descriptor as a linear combination of a subset of \mathbf{x}_i dictionary words = $D \alpha$
$$\mathbf{x}_i = D \alpha$$

Projection local signature => dictionary

- **Sparse coding**
 - We approximate each local descriptor as a linear combination of a subset of x_i dictionary words = $D \alpha$
 $x_i = D \alpha$
 - α weight, D dictionary
 - We force each x_i to be represented only by a small number of visual words => sparsity

$$\alpha_i = \underset{\alpha}{\operatorname{argmin}} L(\alpha, D) \triangleq \|x_i - D\alpha\|_2^2 + \lambda \|\alpha\|_1$$

Projection local signature => dictionary

- **Sparse coding vs VQ: hard assignment standard**
 - SC: Sparse Coding = majority of $\alpha_i = 0$
 - LLC: Local LinearCoding = the words representing the input must be close (locality)
 - Open question: relevance of these minimization criteria on the error of reconstruction in classification ?

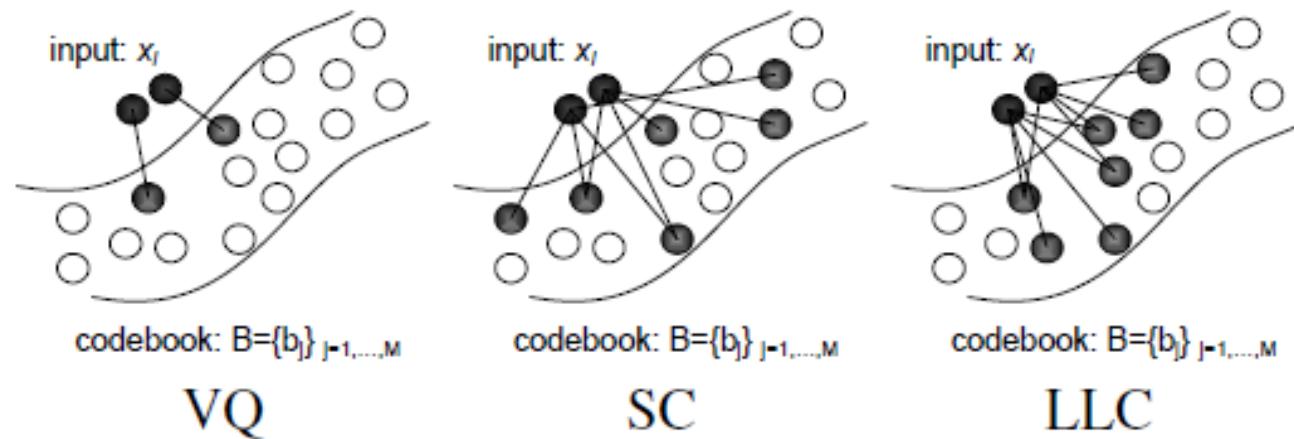


Figure 2. Comparison between VQ, SC and LLC. *The selected bases for representation are highlighted in red*

Pooling: projection aggregation => image global index

$$\mathbf{H} = \begin{matrix} & x_1 & x_j & x_N \\ c_1 & \left[\begin{matrix} \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{matrix} \right] & c_m \\ & \Rightarrow g: \text{pooling} \\ & \Downarrow \\ & f: \text{cooding} \\ c_M & \end{matrix}$$

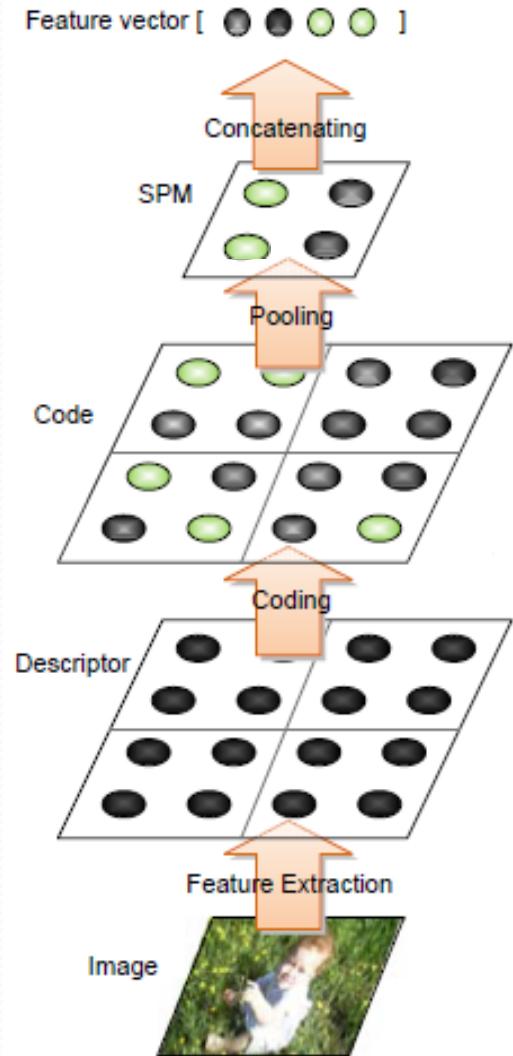
Pooling: projection aggregation => image global index

- Global index: likelihood for the image to contain each visual word
- Different ways to summarize the projections
 - Sum pooling: the most classical BoW approach
 - From text retrieval: count word occurrences in the



Pooling: projection aggregation => image global index

- Global index: likelihood for the image to contain each visual word
- Different ways to summarize the projections
 - Max pooling: we keep the max value of the projection for each visual word
 - Interesting for sparse / soft coding: reduce the impact of the noise
 - Somehow linked to biological inspiration (cortex)



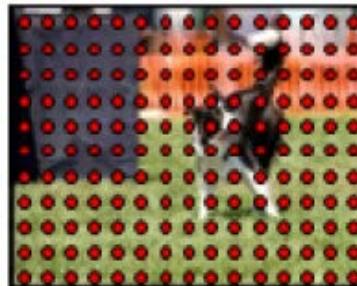
Pooling: projection aggregation => image global index

- BoW: histogram: spatial information about the location of the local descriptors is lost
 - how to store this information?
- Apply the BoW approach in different sub-regions of the image
 - Concatenation of BoWs ?
 - Get a similarity measure combining all levels
 - Spatial Pyramid Matching [LAZEBNIKo6]
 - Extension to the spatial domain of the pyramid match kernel[GRAUMANo5]

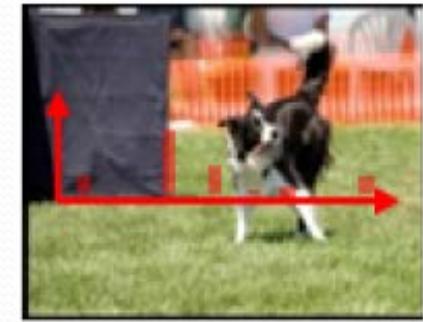
Projection aggregation => image global index



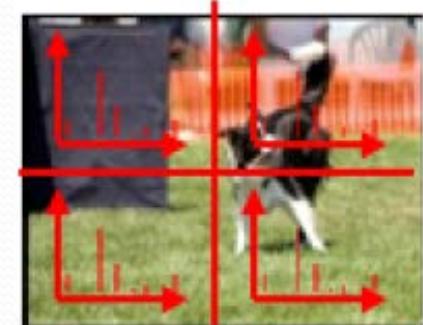
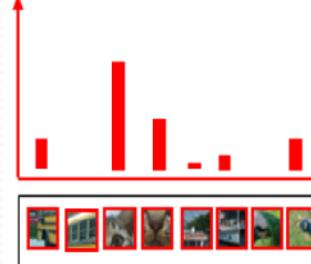
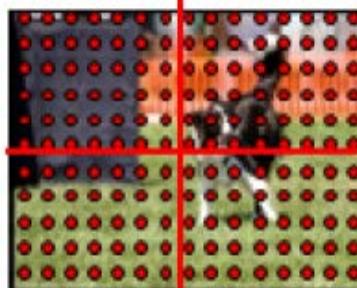
1x1



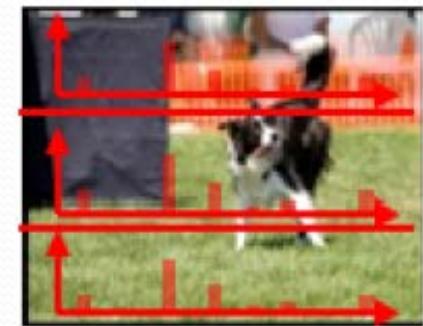
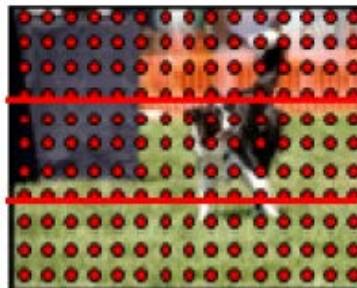
Vocabulary Assignment
(Bag-of-Words model)



2x2



3x1



Two extensions of the BoW

- Spatial Pyramid (Lazebnik, Schmid & Ponce)

Pyramid in image space, quantize features

⇒ Limit the global invariance:

S(  faible

[CVPR 06: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories]

Slide credits: Svetlana Lazebnik

- Pyramid Match Kernel (Grauman & Darrell)

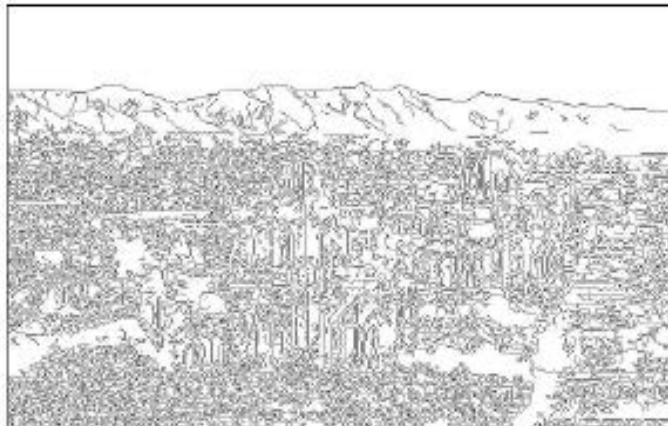
Pyramid in feature space, ignore location

Spatial Pyramid

- Extract interest point descriptors (dense scan)
- Construct visual word dictionary
- Build spatial histograms
- Train an SVM

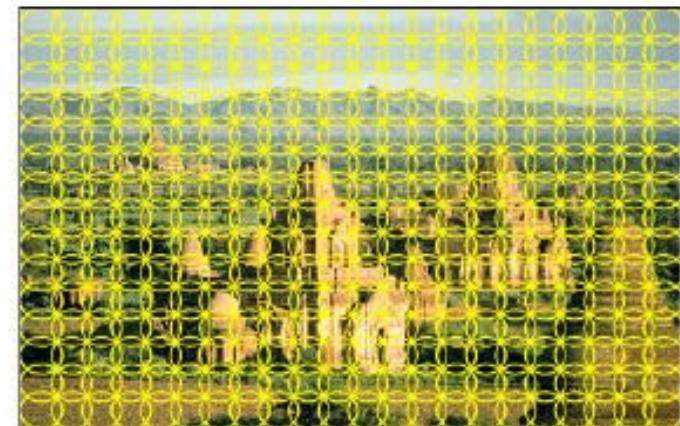
Spatial Pyramid

- Extract interest point descriptors (dense scan)
- Construct visual word dictionary
- Build spatial histograms
- Train an SVM



Weak (edge orientations)

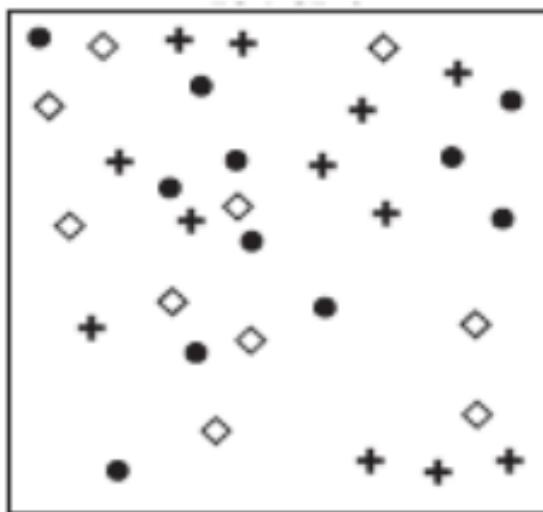
OR



Strong (SIFT)

Spatial Pyramid

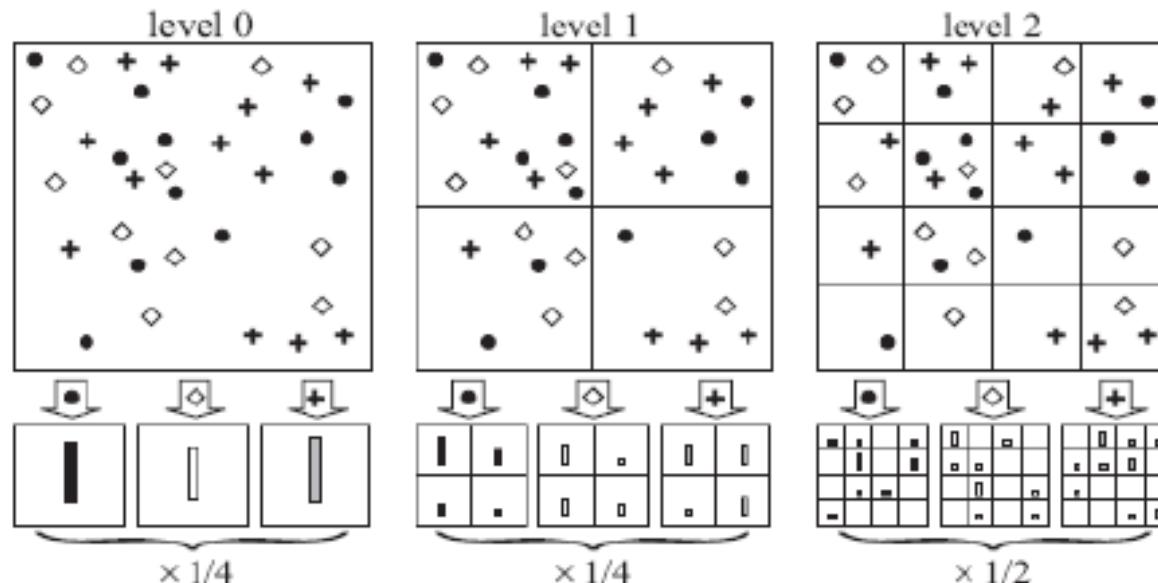
- Extract interest point descriptors (dense scan)
- **Construct visual word dictionary**
- Build spatial histograms
- Train an SVM



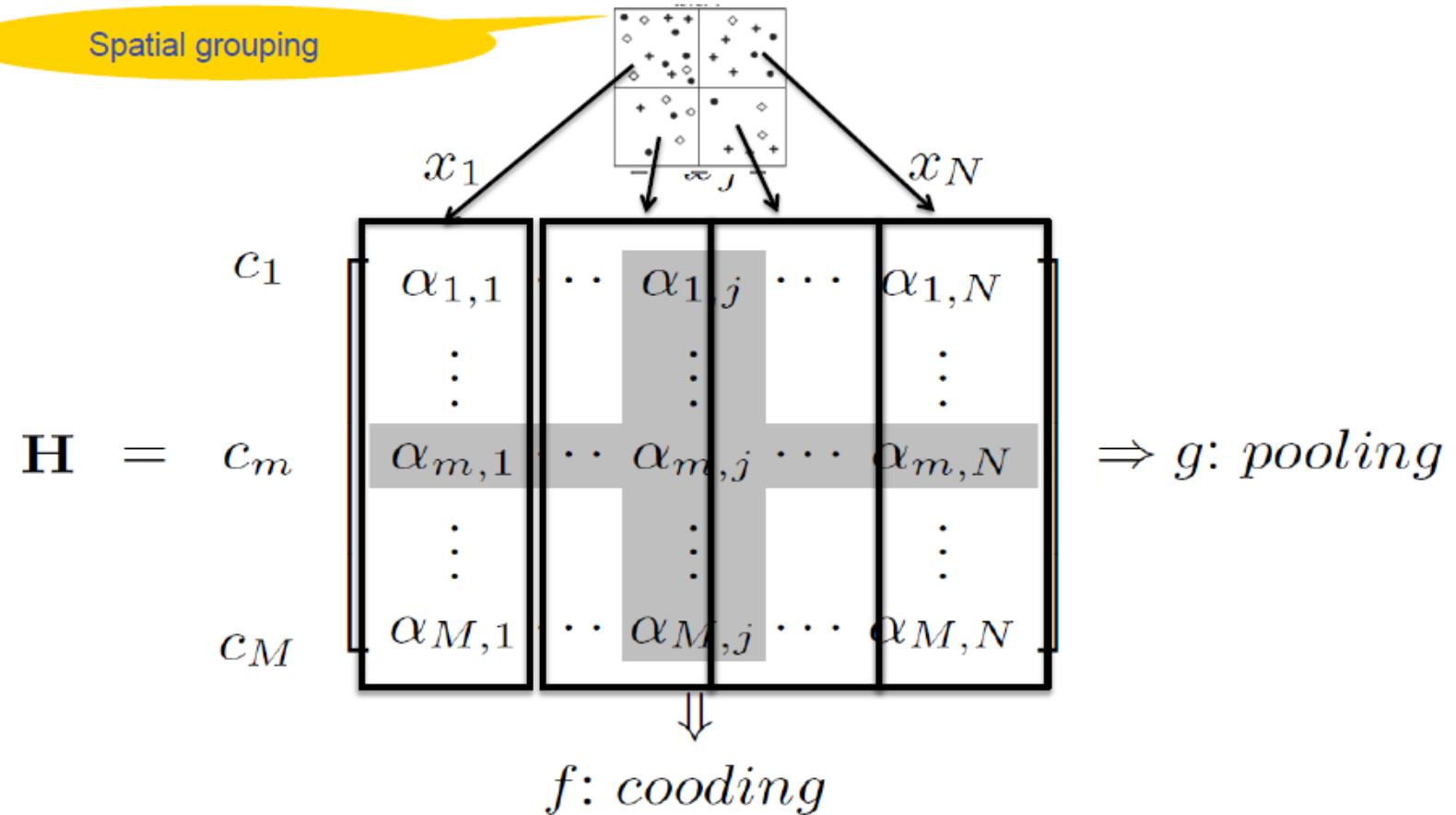
- Vector quantization
- Usually K-means clustering
- Vocabulary size (16 to 400)

Spatial Pyramid

- Extract interest point descriptors (dense scan)
- Construct visual word dictionary
- **Build spatial histograms**
- Train an SVM



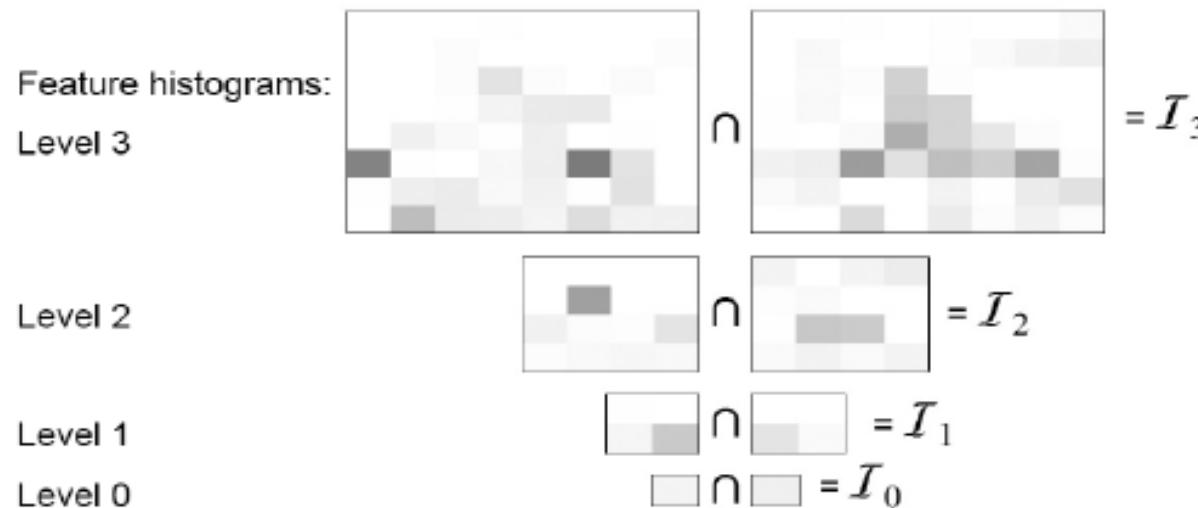
Spatial Pyramid



=> Break the global invariance because the pyramide is fixed

Spatial Pyramid

- Extract interest point descriptors (dense scan)
- Construct visual word dictionary
- Build spatial histograms
- Train an SVM



Total weight (value of *pyramid match kernel*): $\mathcal{I}_3 + \frac{1}{2}(\mathcal{I}_2 - \mathcal{I}_3) + \frac{1}{4}(\mathcal{I}_1 - \mathcal{I}_2) + \frac{1}{8}(\mathcal{I}_0 - \mathcal{I}_1)$

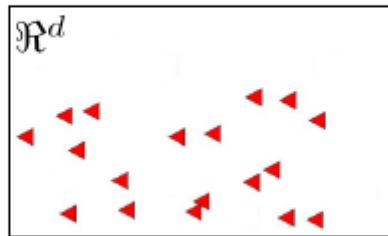
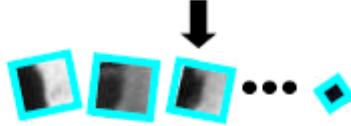
Two extension of BoW

- Spatial Pyramid (Lazebnik)
*Pyramid in **image** space, quantize features*

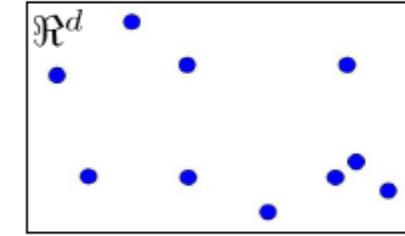
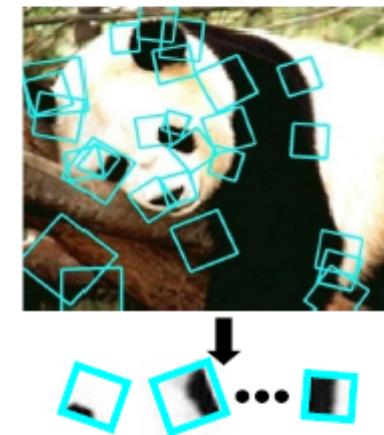
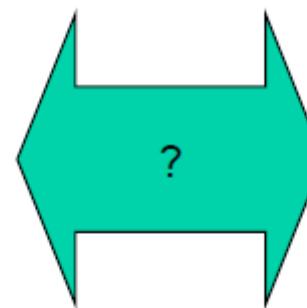
- **Pyramid Match Kernel (Grauman & Darrell)**
*Pyramid in **feature** space, ignore **location**
=> Idée de “corriger” les **imperfections** du dico*

How to compare Sets of Features?

- Each instance is unordered set of vectors
- Varying number of vectors per instance

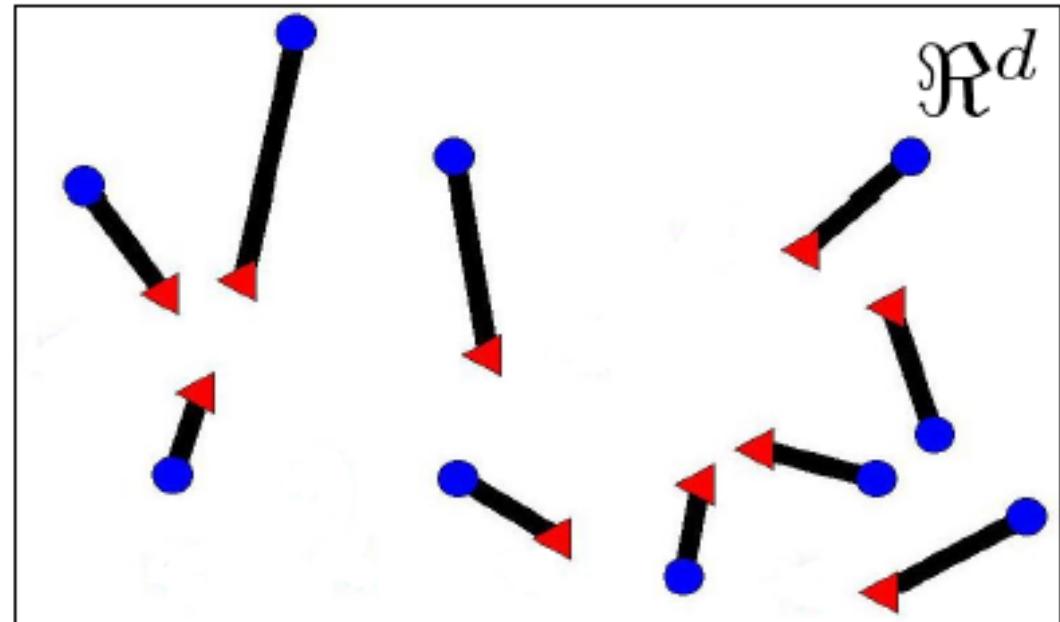


$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \dots, \vec{\mathbf{x}}_m\}$$



$$\mathbf{Y} = \{\vec{\mathbf{y}}_1, \dots, \vec{\mathbf{y}}_n\}$$

Correspondence-based Match

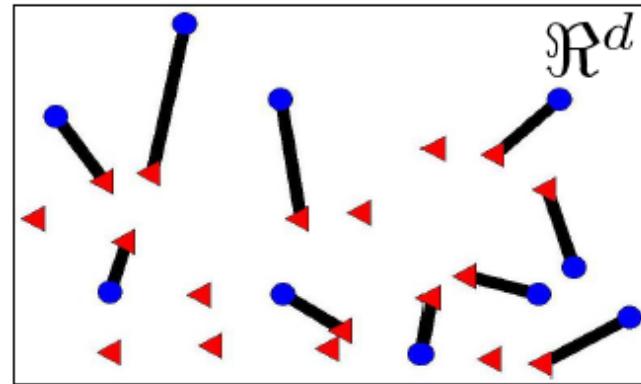


Explicit search for correspondences...

$$\max_{\pi: \mathbf{X} \rightarrow \mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

Partial Matching

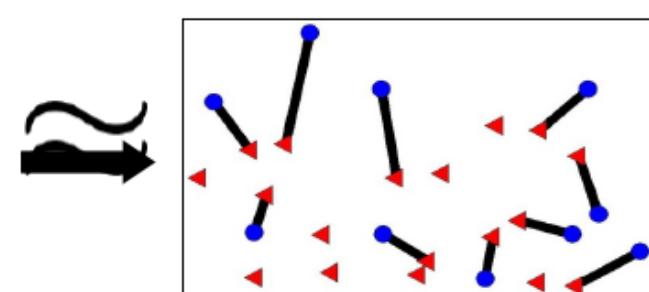
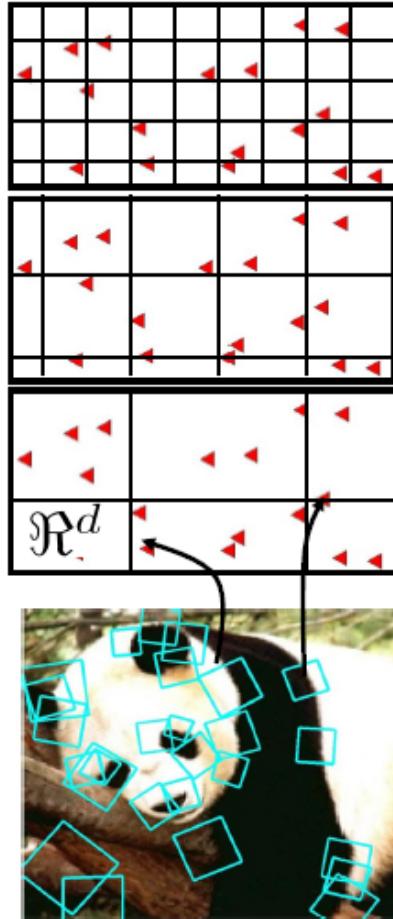
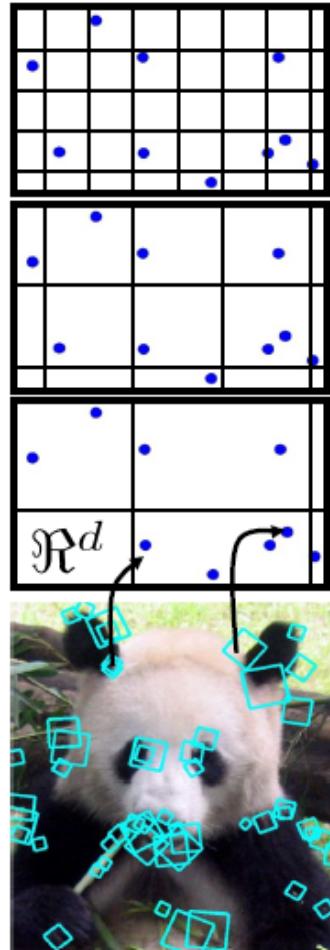
Compare sets by computing a *partial matching* between their features.



$$\max_{\pi: \mathbf{X} \rightarrow \mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$



Pyramid Matching



optimal partial
matching

$$\max_{\pi: \mathbf{X} \rightarrow \mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \dots, \vec{\mathbf{x}}_m\} \quad \mathbf{Y} = \{\vec{\mathbf{y}}_1, \dots, \vec{\mathbf{y}}_n\}$$

Pyramid Matching

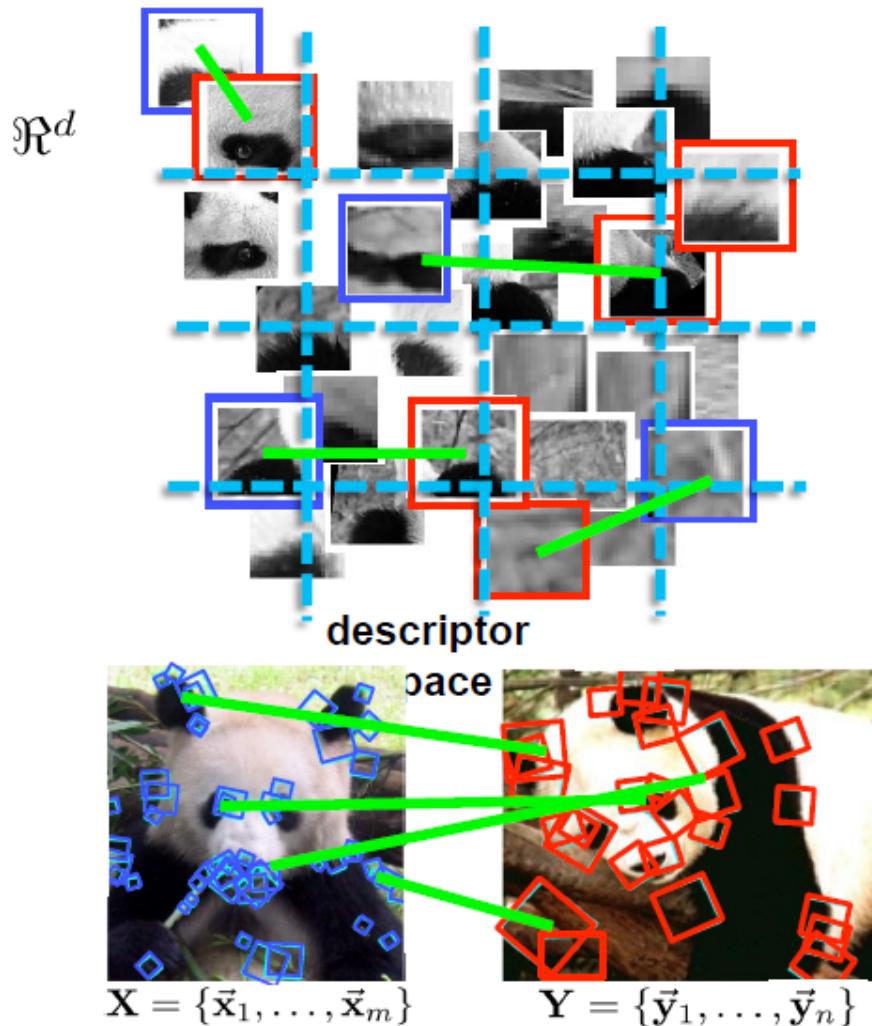
$$\mathbf{H} = \begin{matrix} & \text{---} \\ c_1 & \left[\begin{array}{cccc} \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{array} \right] \\ c_m & \text{---} \\ c_M & \text{---} \end{matrix} \Rightarrow g: \text{pooling}$$

↓

f: cooding

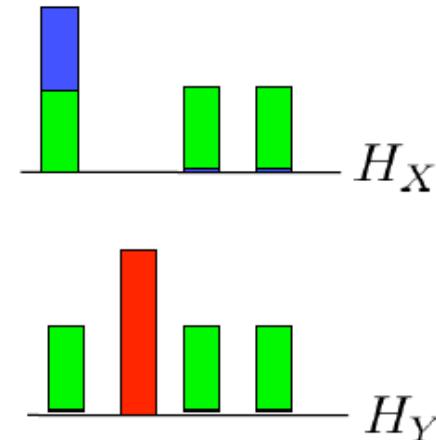
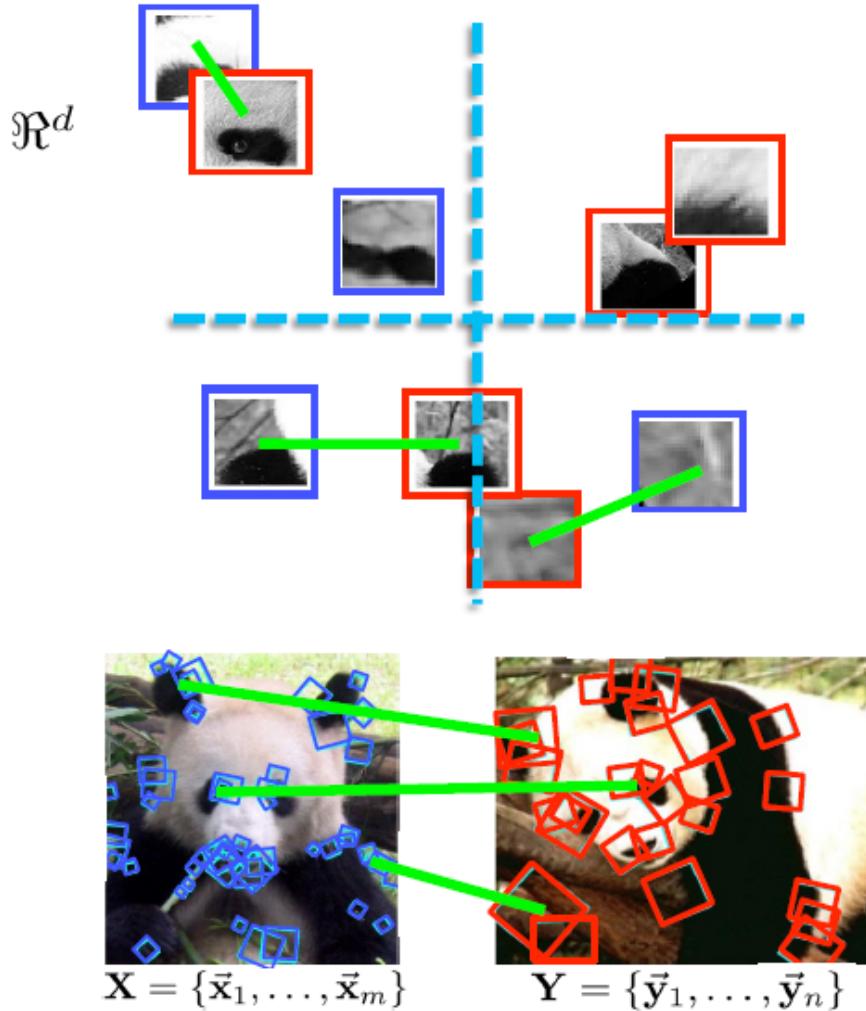
Work on dictionary

Pyramid Matching: Main Idea



Feature space partitions serve to “match” the local descriptors within successively wider regions.

Pyramid Matching: Main Idea



$$\begin{aligned}\mathcal{I}(H_X, H_Y) &= \sum_j \min(H_X(j), H_Y(j)) \\ &= 3\end{aligned}$$

Histogram intersection counts number of possible matches at a given partitioning.

Pyramid Match Kernel

$$K_{\Delta}(X, Y) = \sum_{i=0}^L 2^{-i} \mathcal{I}\left(H_X^{(i)}, H_Y^{(i)}\right) - \mathcal{I}\left(H_X^{(i-1)}, H_Y^{(i-1)}\right)$$

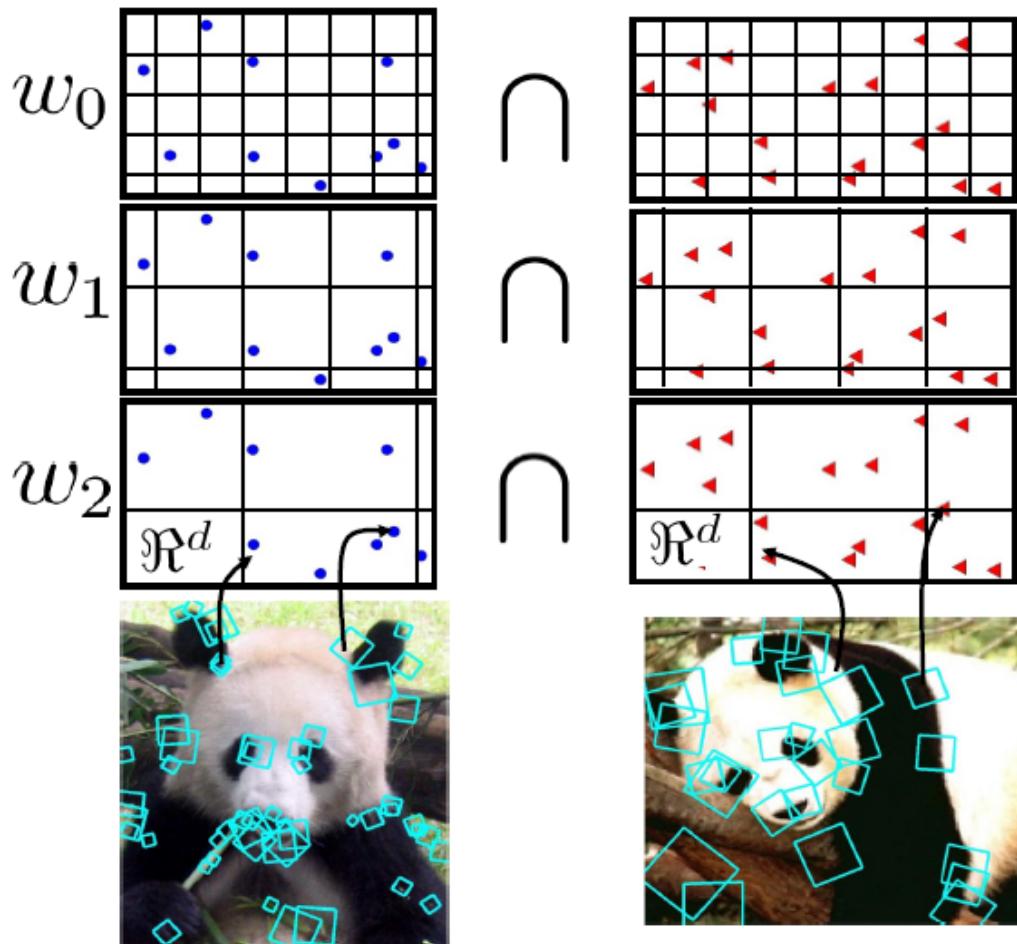
measures
difficulty of a
match at level i

number of newly matched pairs
at level i

- For similarity, weights inversely proportional to bin size (or may be learned)
- Normalize these kernel values to avoid favoring large sets

[Grauman & Darrell, ICCV 2005]

Pyramid Match Kernel



$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \dots, \vec{\mathbf{x}}_m\} \quad \mathbf{Y} = \{\vec{\mathbf{y}}_1, \dots, \vec{\mathbf{y}}_n\}$$

Optimal match: $O(m^3)$
Pyramid match: $O(mL)$

