Recognition/classification

1. Introduction
2. Supervised learning
3. SVM classifiers
4. **Datasets and evaluation**

Datasets for learning/testing

- How to define a category ?
  - Bicycle
  - Paintings with women
  - Portraits

  …

  Concepts, semantics, ontologies …

Image/video datasets for training/testing

CalTech 101

TRECVID

- Choice of the categories (objects, concepts)
  - Number of categories
  - Number of images per category
  - Hierarchical structure ?
- Mono-label/multi-labels
- Pre-processing
  - Color, resolution, centered …

Example: ImageNet dataset

- Large Scale Visual Recognition Challenge (ILSVRC)
  - 1,2 Million images, 1000 classes

- Paper:
  - ImageNet: A Large-Scale Hierarchical Image Database, Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei, CVPR 2009

Classes of ImageNet

- Based on WordNet
  - Each node is depicted by images
- A knowledge ontology
  - Taxonomy
  - Partonomy

- Website:

IMAGENET

14,197,122 images, 21841 synsets indexed

Explore Download Challenges Publications CoolStuff About

Not logged in. Login I Signup

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

Click here to learn more about ImageNet, Click here to join the ImageNet mailing list.

Constructing ImageNet

- 2-step process

Step 1 :
Collect candidate
images Via the Internet

Step 2 :
Clean up candidate
Images by humans

Step 1: Collect Candidate Images from the Internet

- For each synset, the queries are the set of WordNet synonyms
- Accuracy of Internet Image search results: 10 % (in 2010)
  - For 500-1000 clean images, needs 10K images
- Query expansion
  - Synonyms: German police dog, German shepherd dog
  - Appending words form ancestors: sheepdog, dog
- Multiple Languages
  - Italian, Dutch, Spanish, Chinese …
- More engines: Yahoo! , flickr, Google
- Parallel downloading

Step 2: Clean up the candidate Images by humans

- Rely on humans to verify each candidate image collected for a given synset
- Amazon Mechanical Turk (AMT)
  - Present the users with a set of candidate images and the definition of the target synset
  - let users select the best match ones

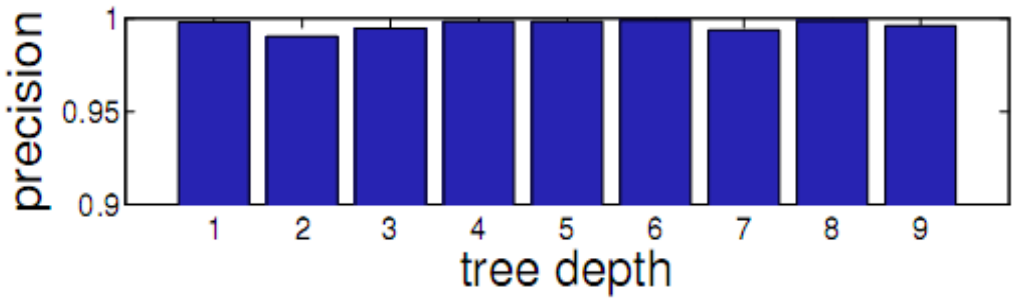
Ensure Accuracy

- Users Enhancement
  - Provide wiki and links for definitions
  - Make sure workers read the definition
    - Definition quiz
- Human users make mistakes (3% wrong annotations)
- Not all users follow the instructions
- Users do not always agree with each other
  - Subtle or confusing synsets, e.g. Burmese cat
- Quality Control System
  - Randomly sample an initial subset of images to users
    - Have multiple users independently label same image
  - Obtain a confidence score table, indicating the probability of an image being a good image given the user votes
    - Different categories requires different levels of consensus
  - Proceed until a pre-determined confidence score threshold reached

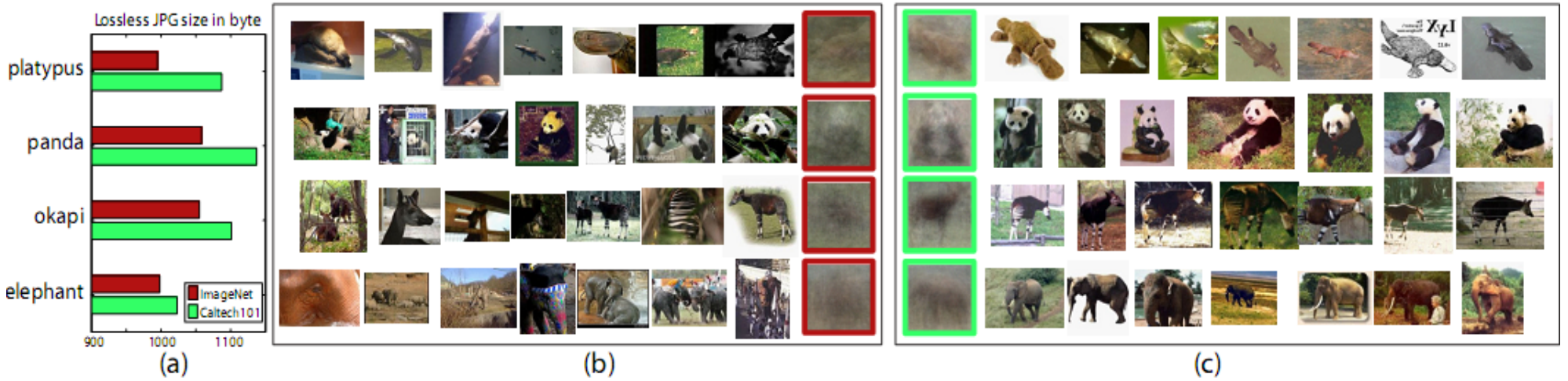
User 1	Y	Y	Y
User 2	N	Y	Y
User 3	N	Y	Y
User 4	Y	N	Y
User 5	Y	Y	Y
User 6	N	N	Y

Properties of ImageNet

Accuracy:
clean dataset at all level



Diversity:
variable appearances, positions, view points, poses, background clutter, occlusions.



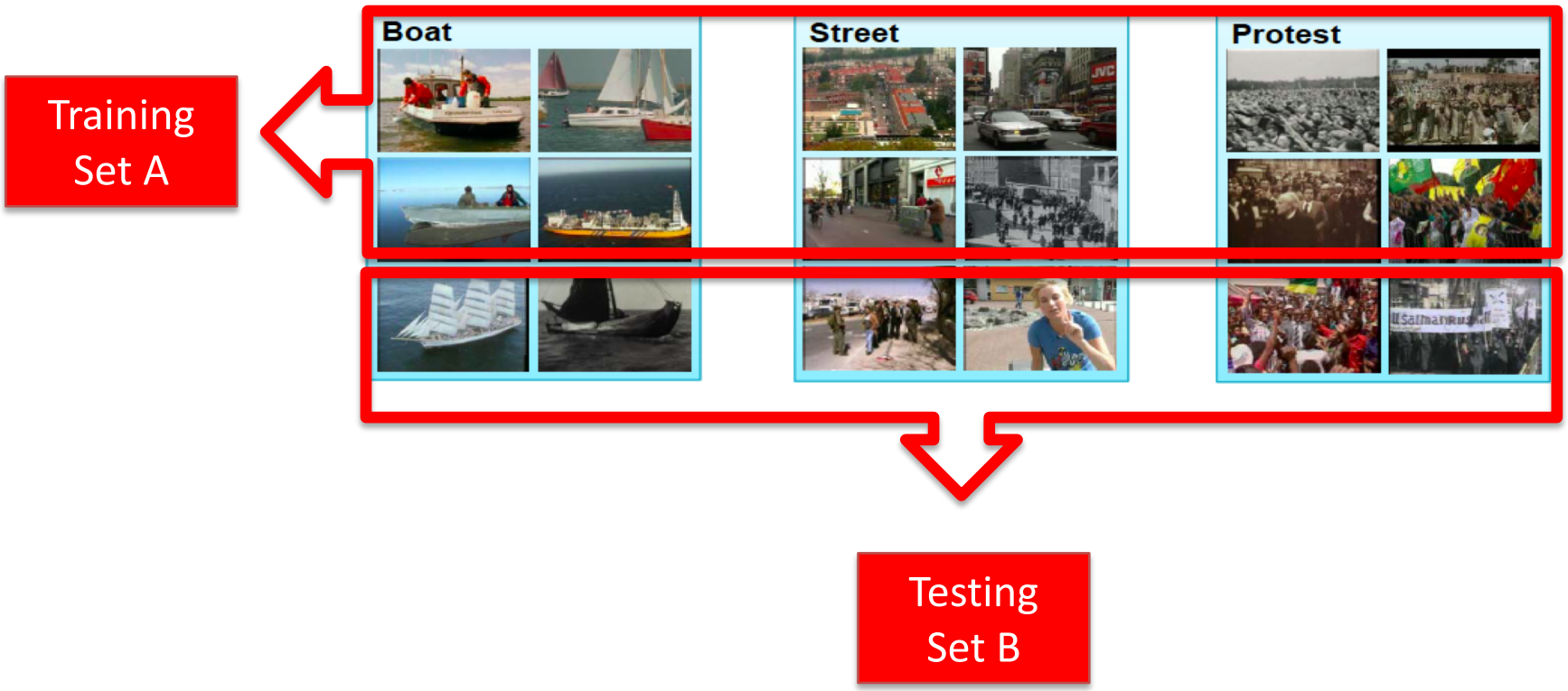
Limitations:

- Crawling bias, Improve algorithm: PageRank
- AMT: hierarchical users based on their ability
- Only one tag per image

Benchmarks and evaluation

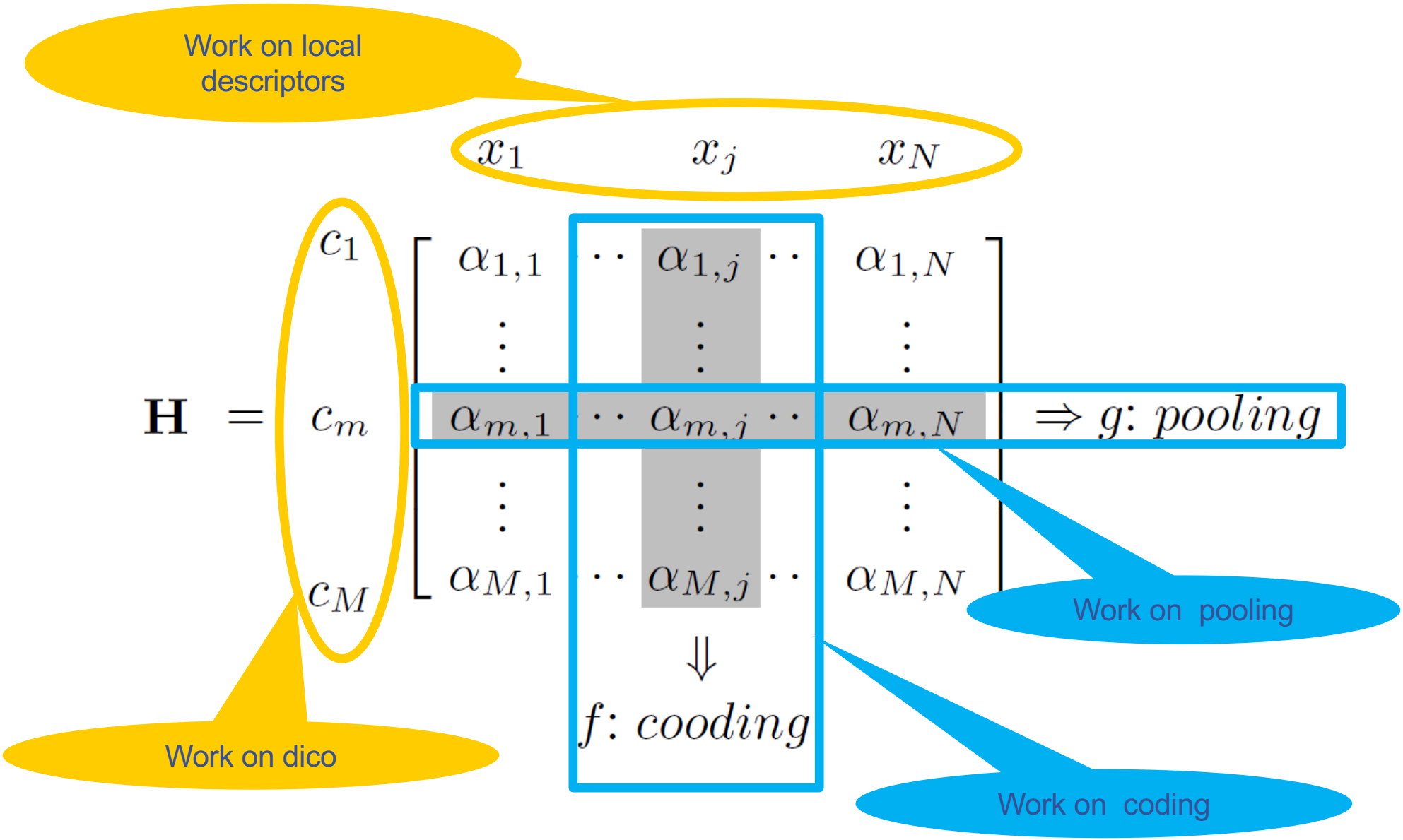
- Train / test / validation sets
  - Cross-validation
  - Learning hyper parameters
- Evaluation
  - Test Error
  - Accuracy, MAP, confusion matrix, Per-class averaging
  - Significance of the comparison, statistical tests, …
- Dataset building, concepts and semantics
  - Data pre-processing, data augmentation

Image/video datasets for training/testing



- Training classifiers on A
- Testing on B: error evaluation
- A and B disjoints!

Beyond BoW representation


Work on local descriptors
x_1 x_j x_N
$\mathbf{H} =$
c_1 c_m c_M
$\alpha_{1,1}$ $\alpha_{1,j}$ $\alpha_{1,N}$
$\alpha_{m,1}$ $\alpha_{m,j}$ $\alpha_{m,N}$
$\alpha_{M,1}$ $\alpha_{M,j}$ $\alpha_{M,N}$
$\Rightarrow g$: pooling
$\Downarrow$
f: cooding
Work on dico
Work on pooling
Work on coding

Pooling: Aggregating projections => global image index

Sum pooling alternative:

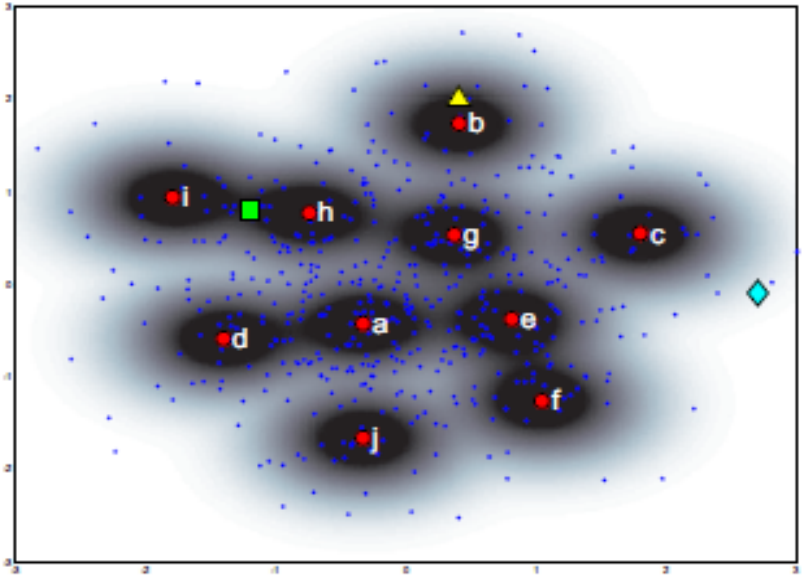
- **Max pooling** : keep the max value for the projection for each visual word
  - Relevant for sparse / soft coding: limit noise effect
  - (Partially) Justify by bio-inspired models (cortex)

$$z_m = g(\alpha_m) = \max_{j=1..N} \alpha_{m,j}$$

$$\mathbf{H} = \begin{array}{c} \\ c_1 \\ \\ c_m \\ \\ c_M \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & & x_j & & x_N \end{array} \\ \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{bmatrix} \\ \Downarrow \\ f\text{: cooding} \end{array} \Rightarrow g\text{: pooling}$$

Coding: Projection =>dictionary

- **soft assignment**
  - Kernel codebook : absolute weight
  - Uncertainty: relative weight
  - Plausibility: absolute weight to 1-nn



Visual Word Ambiguity

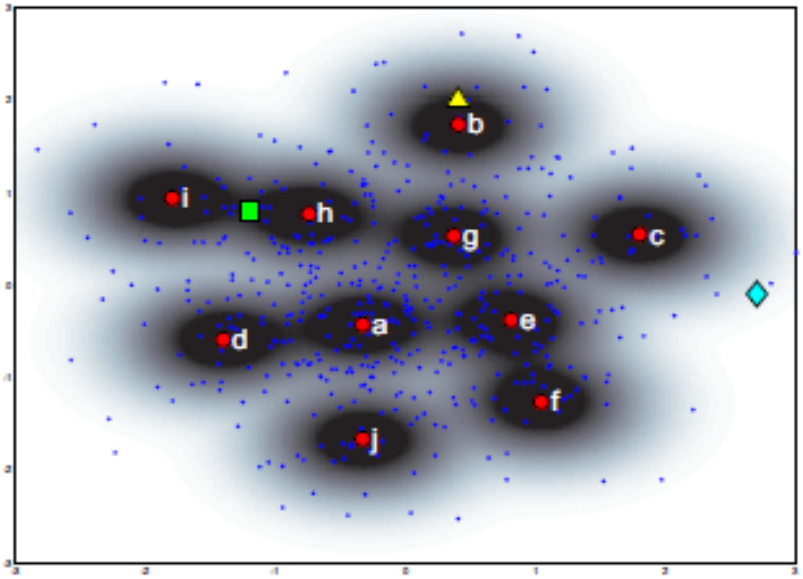
J.C. van Gemert, C.J. Veenman, A.W.M. Smeulders, J.M. Geusebroek

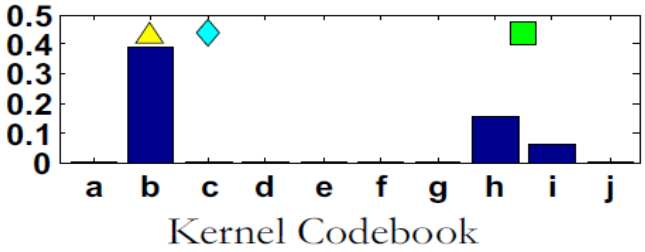
PAMI 2010

Soft Coding :kernel

$$f_{Kernel}(x_j)[m] = K(d(x_j, c_m))$$

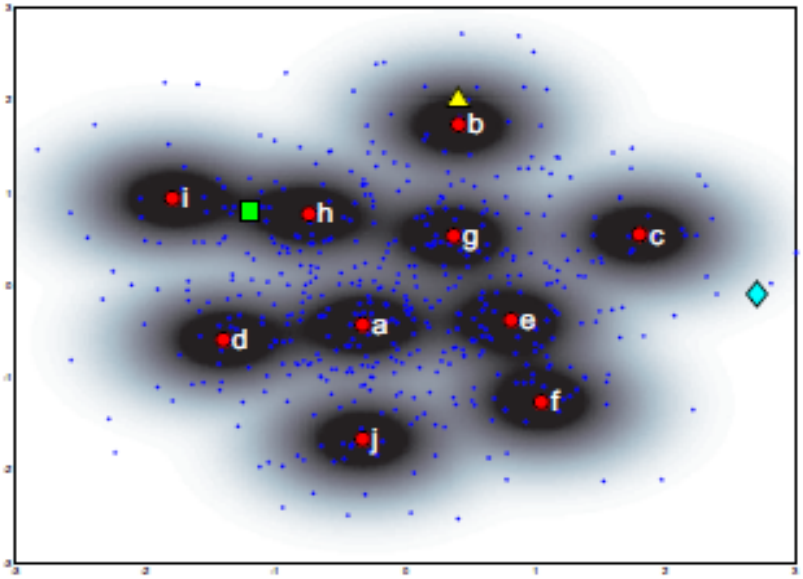
Ex: K(x)=exp(-x)

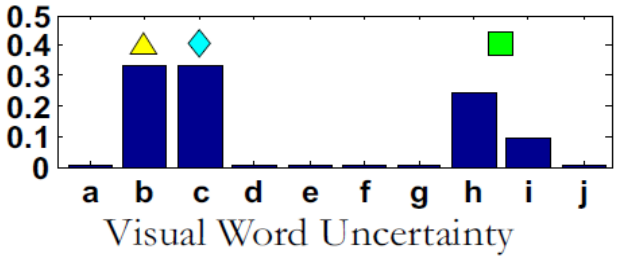




Soft Coding : uncertainty

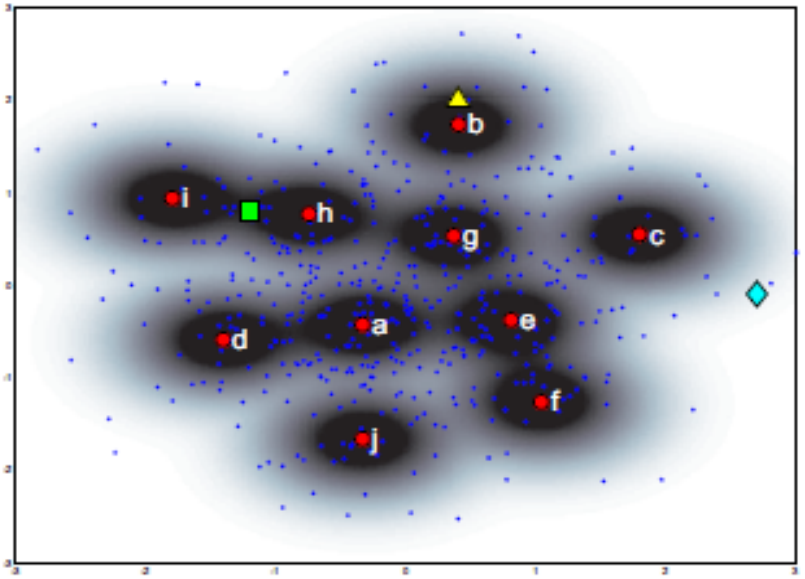
$$f_{Unc}(x_j)[m] = \frac{K(d(x_j, c_m))}{\sum_{k=1}^{M} K(d(x_j, c_k))}$$

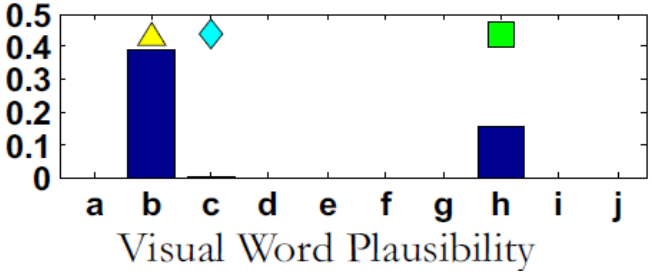




Soft Coding : plausibility

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

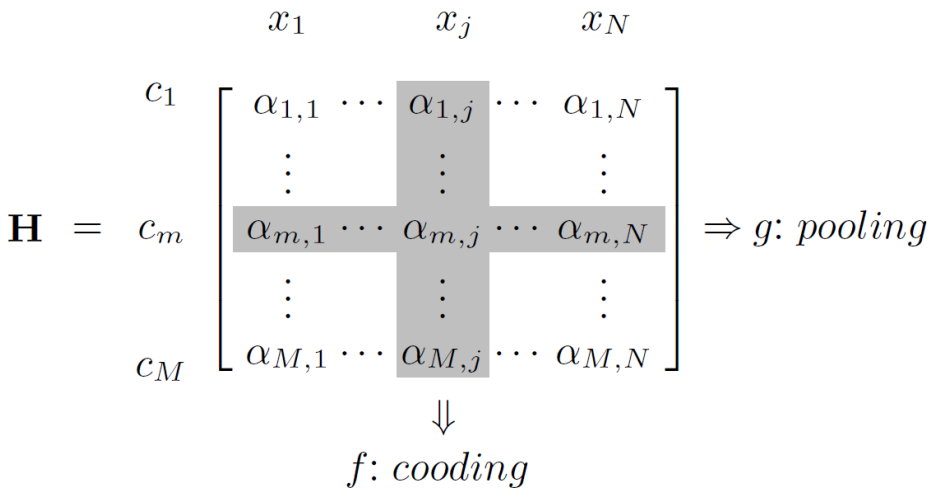




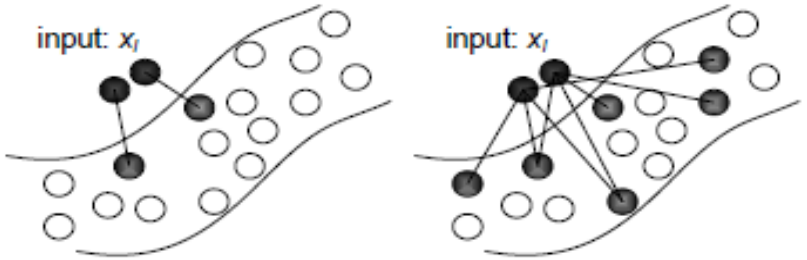
Soft Coding

- Soft vs hard assignement/coding
  - Not a so big gain soft / hard
  - Uncertainty certainly the best strategy

- Semi-soft : excellent tradeoff



Sparse Coding



- Other approach: **sparse coding**
  - Approximation of each local feature x_i (SIFT) as a lin. combination of a subset of words from the dictionary: $x_i \sim C\alpha_i$
    - α_i weight vectors, C matrix of vectors of the dictionary

$$\alpha_i = \underset{\alpha}{argmin} \quad L(\alpha, C) \triangleq \|x_i - C\alpha\|_2^2$$

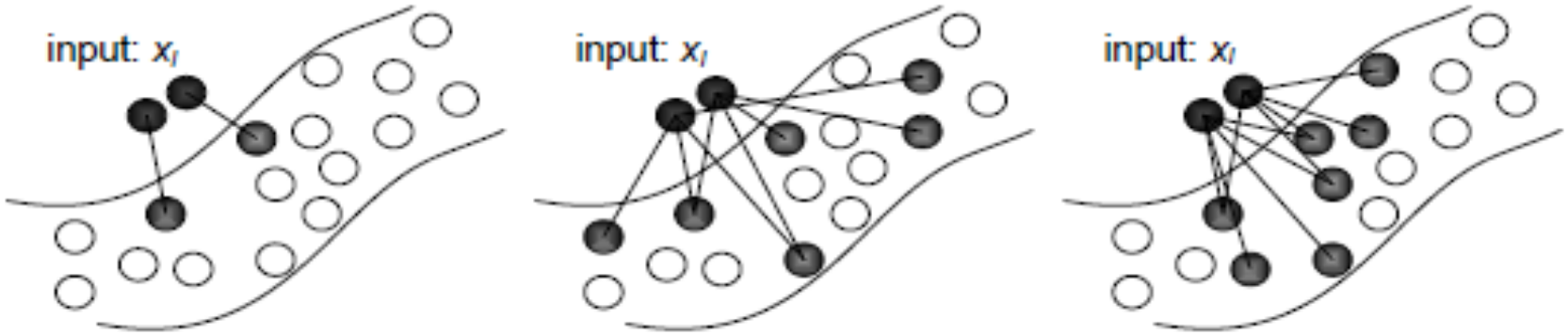
    - Pb: not sparse, many irrelevant values in M
    - Each x_i should be represented using only a small nb of visual words => sparsity

  - Sparse but no locality

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

Sparse Coding

- Sparse coding vs VQ (hard assignment)
  - VQ: hard coding
  - SC : Sparse Coding : most of α_i=0
  - LLC : Local Linear Coding : words representing the feature must be close (locality)



VQ SC LLC

- Are these criteria minimizing reconstruction error relevant for image classification purpose?

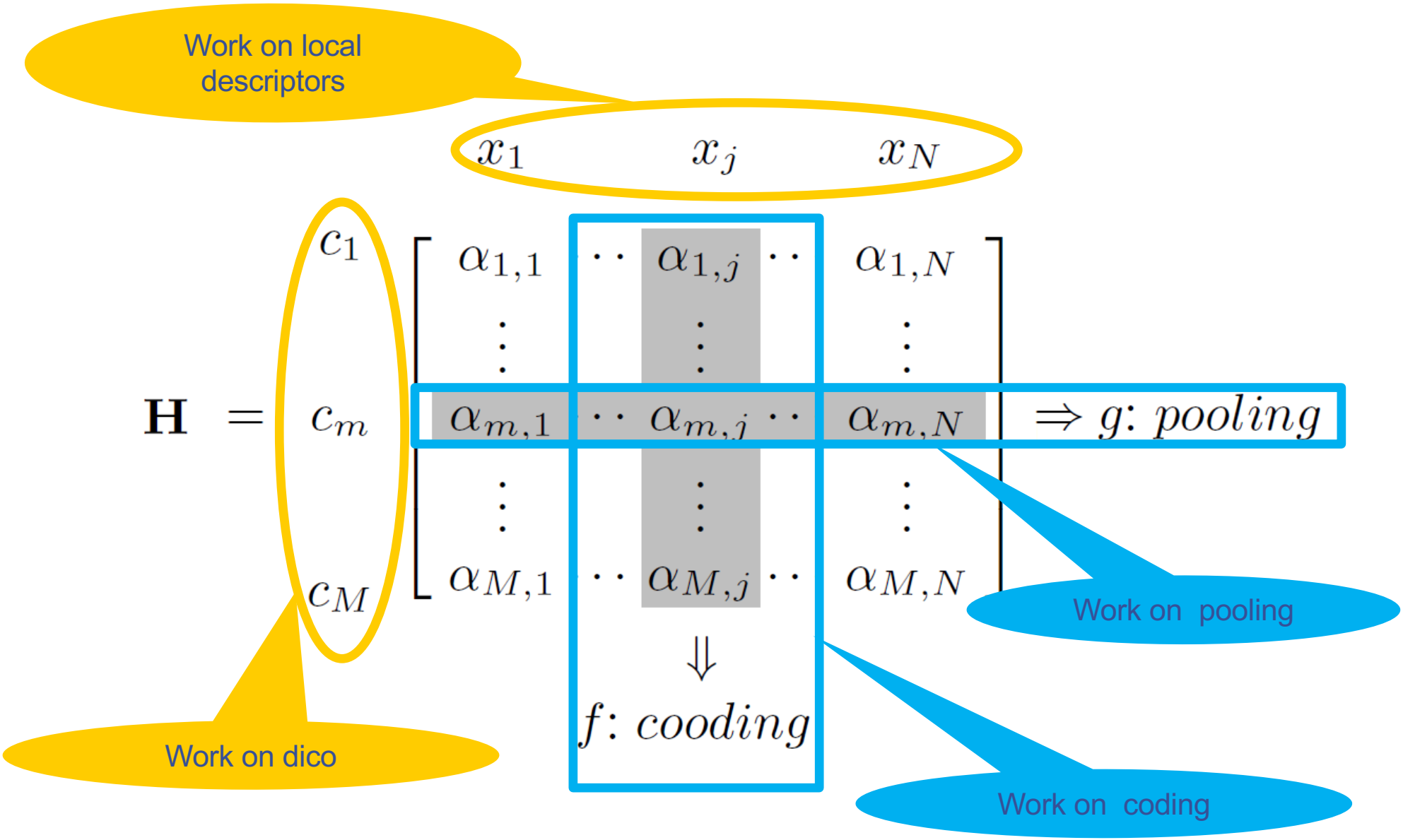
Aggregating projections => global image index

Where we are:

- Better represent coding/pooling association: work on the whole matrix of clusters/descriptors dependency => combine spatial pooling and sparse coding

Next:

- Work on new descriptors (bio inspired, learned)
- Dictionaries
  - Train the dico (supervised training)
  - Avoiding dico/clustering
    - Kernel similarity on bag of local features
- Exploit spatial image information

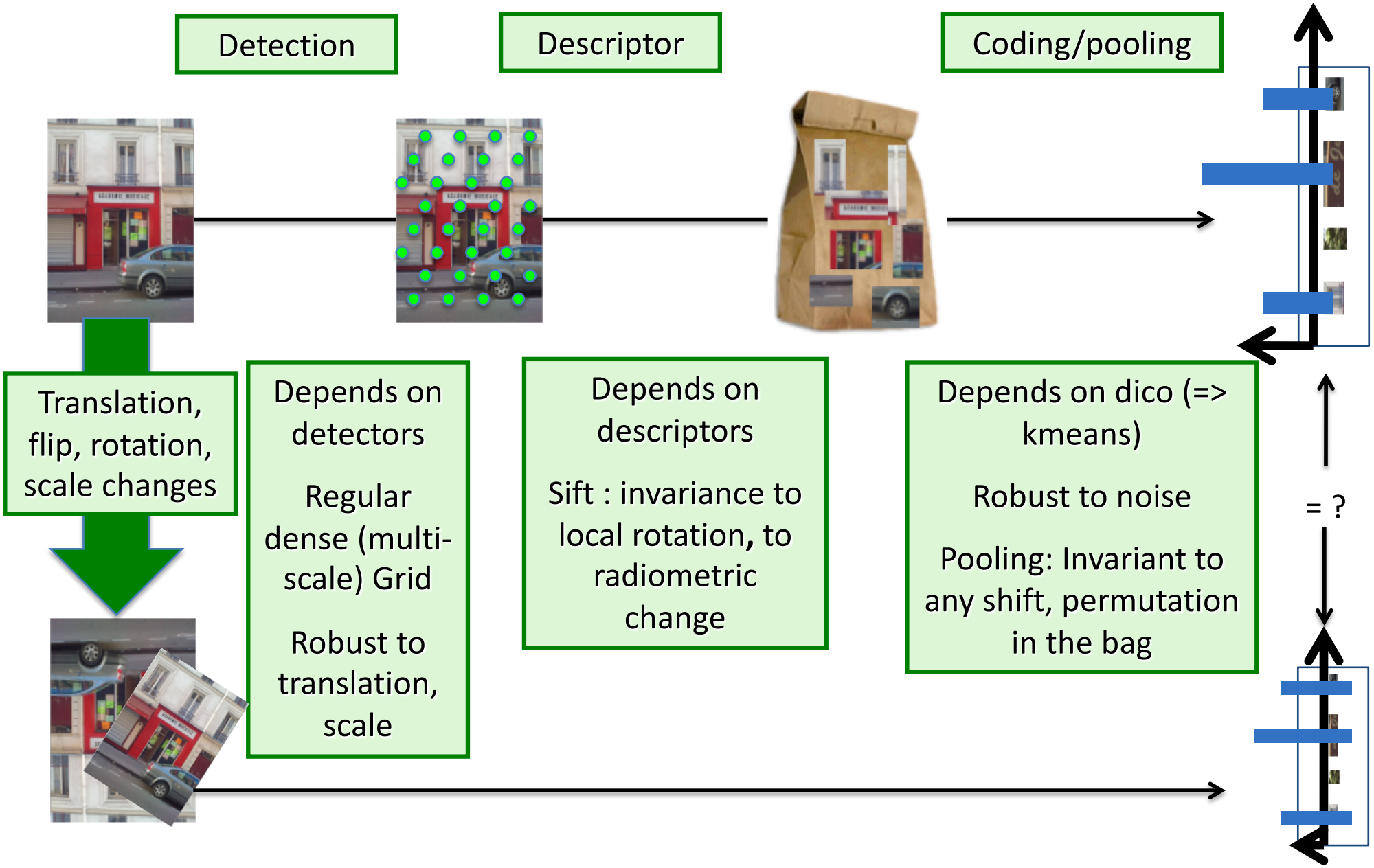

Work on local descriptors
x_1 x_j x_N
$\mathbf{H} =$
c_1
c_m
c_M
$\alpha_{1,1}$ $\alpha_{1,j}$ $\alpha_{1,N}$
$\alpha_{m,1}$ $\alpha_{m,j}$ $\alpha_{m,N}$
$\alpha_{M,1}$ $\alpha_{M,j}$ $\alpha_{M,N}$
$\Rightarrow g$: pooling
$\Downarrow$
f: cooding
Work on pooling
Work on coding
Work on dico

Invariance/robustness in BoW pipeline

Stability of the representation:

- Small deformations/transformations in the input space => similar representations
- Large (or unexpected) transformations in the input space => very dissimilar representations

Invariance/robustness in BoW pipeline



Beyond BoW

- **SPM: Spatial Pyramid (Lazebnik et al)**

  Geometry in BoW: Pyramid in image space

- Pooling (Avila et al)

  Advanced representation for pooling

- Pyramid Match Kernel (Grauman et al)

  Pyramid in feature space: Kernel similarity

Article analysis

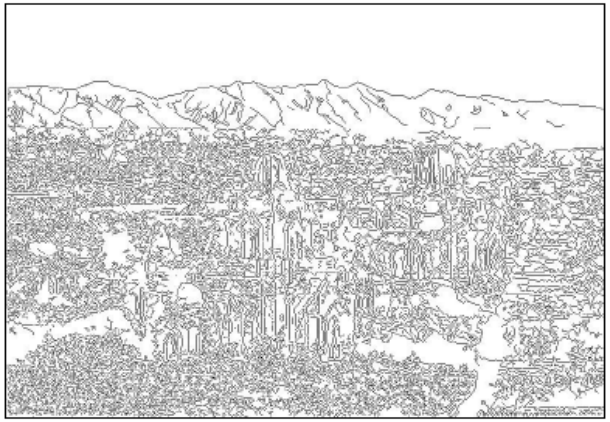
Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, Lazebnik et al, CVPR 06

1. SPM algorithm: is it a BoW model?
2. Which descriptors (type, size, …)?
   - extraction scheme (dense/sparse, sampling, … ?)
   -details about filtering of local features
3. Which technique to construct visual word dictionary?
4. **How spatial info is introduced (SPM)? Motivation ?**
   -Detail of the algo
   -Final size of the representation
5. Which classification algo?
6. Results

   Comments on tables and fig.

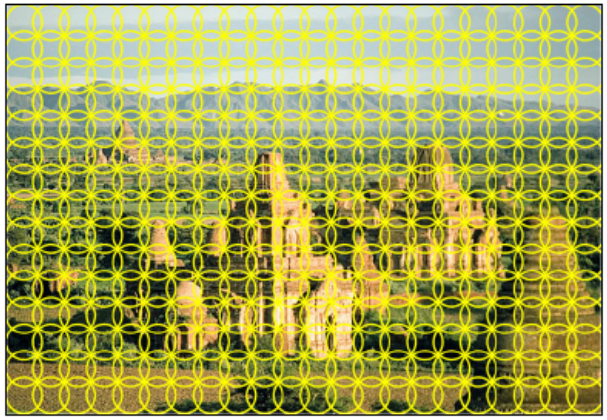
SPM Algorithm

1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. Train an SVM



Weak (edge orientations)

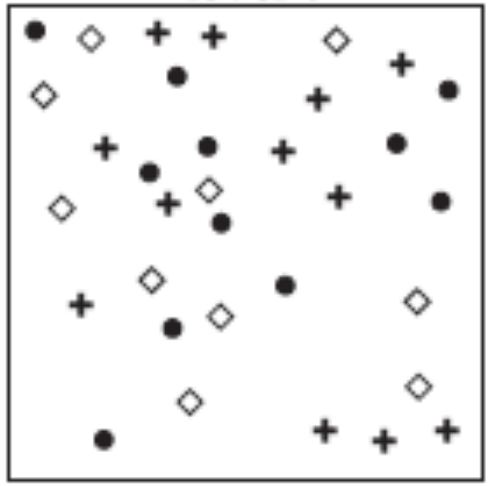
OR



Strong (SIFT)

Algorithm

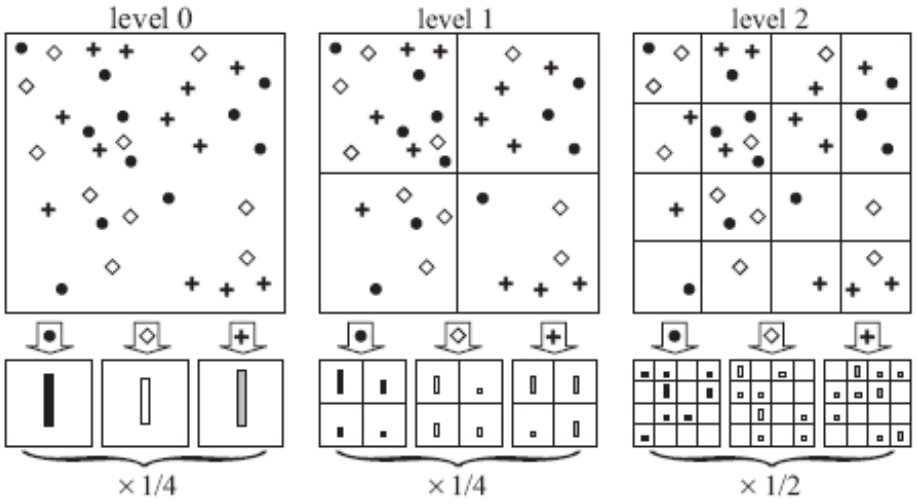
1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. Train an SVM

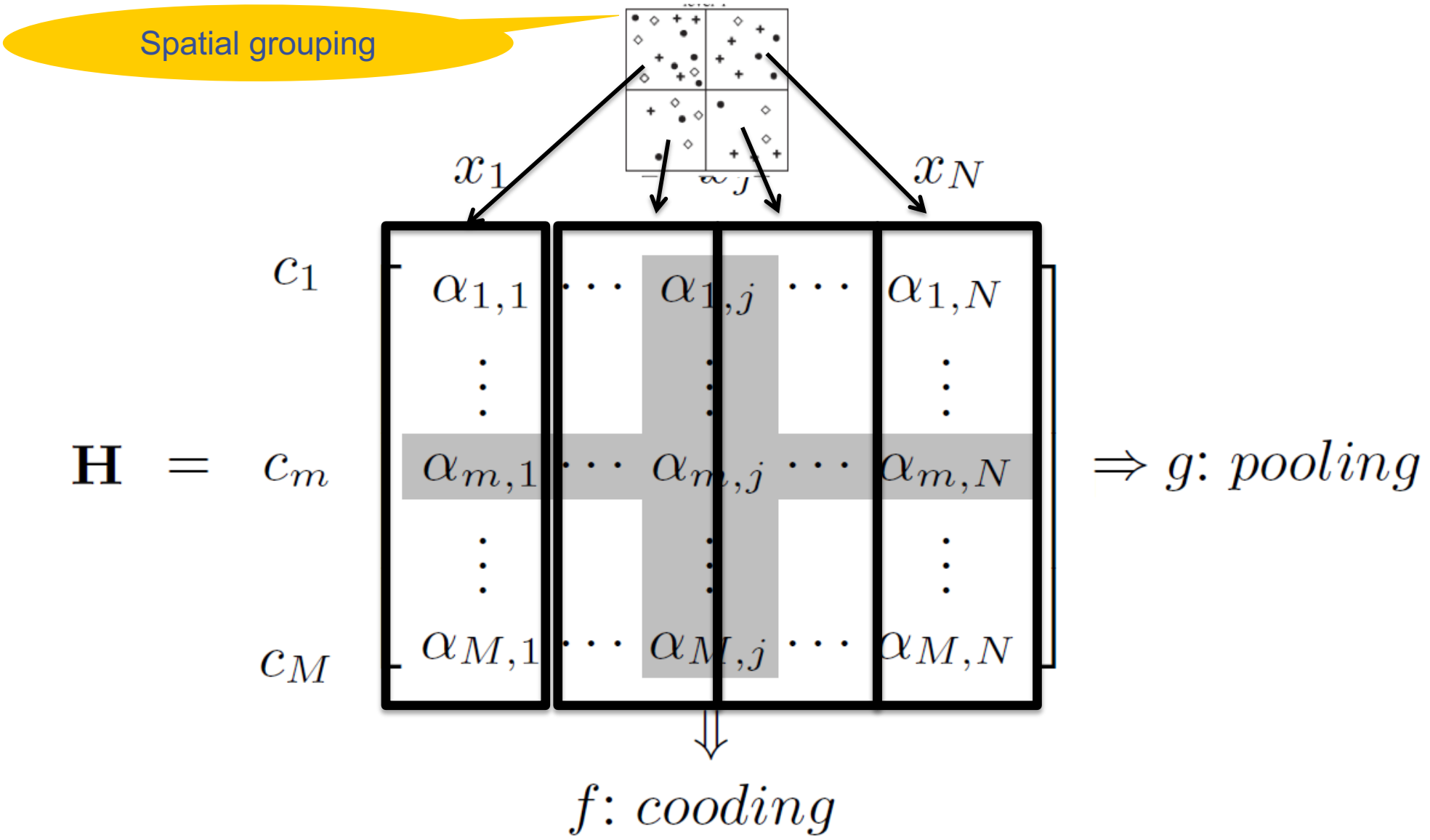


- Vector quantization
- Usually K-means clustering
- Vocabulary size (16 to 400)

Algorithm

1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. Train an SVM (with specific kernels)





=> Break global invariance because of fixed pyramid

- ***Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, Lazebnik et al, CVPR 06***

  Pyramid in image space, quantize features

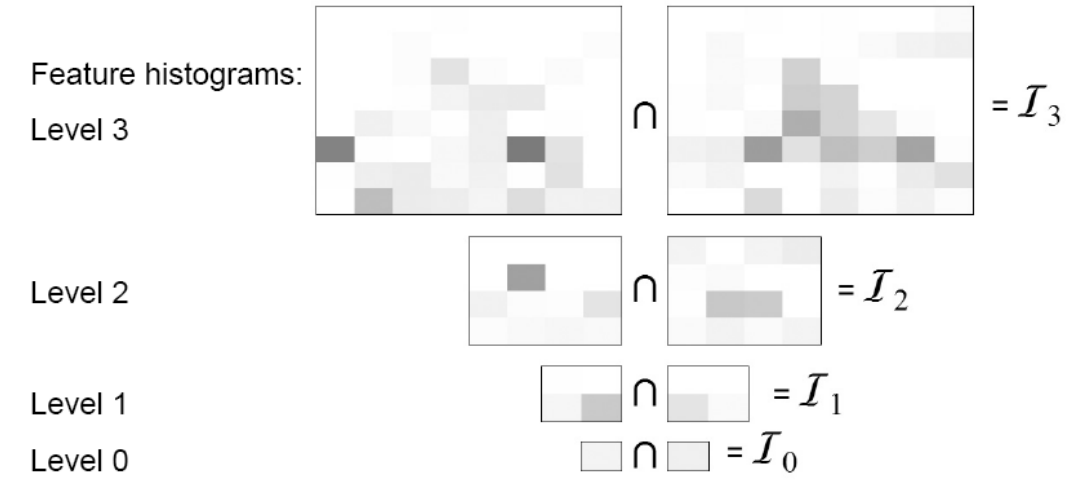
  ⇒**Limit the global invariance:**

  S( , ) small

Algorithm

1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. Train an SVM

Similarity



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

Algorithm

1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. **Train an SVM**

… Based on the kernel Similarity PMK