# Kernel Codebooks for Scene Categorization

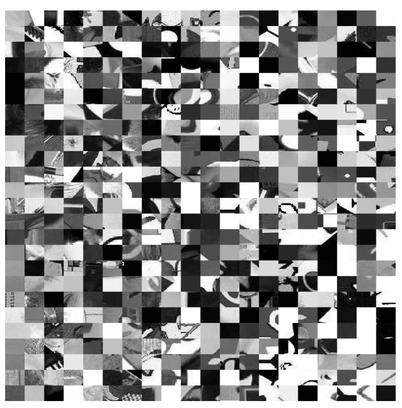
**Jan van Gemert**[1,2] **Jan-Mark Geusebroek**[1] **Cor Veenman**[1] **Arnold Smeulders**[1]

1: ISLA, University of Amsterdam

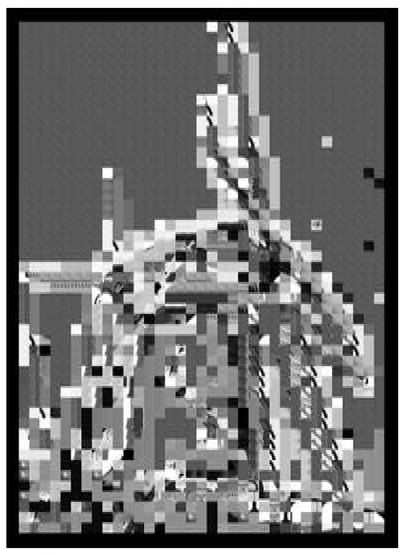
2: Willow, École Normale Supérieure

## 1. Codebooks for Scene Categorization

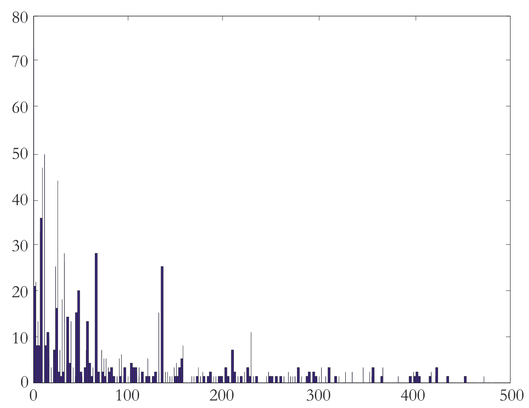
Original image



Prototype vocabulary



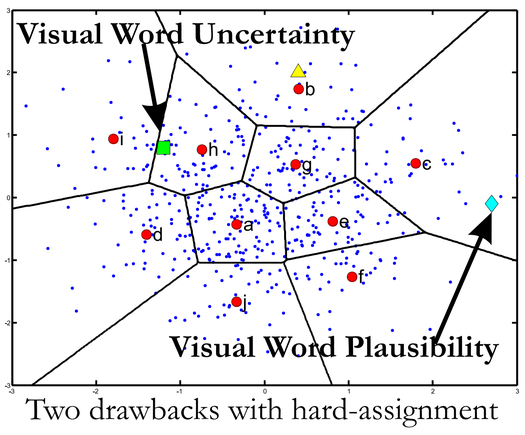
Feature to best prototype




Prototype frequency histogram

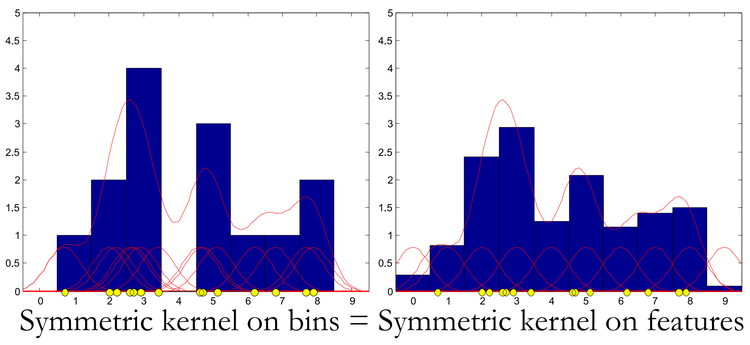
- Images as bag-of-features
- A feature represented by a discrete prototype in the vocabulary
- A vocabulary is commonly created by unsupervised clustering
- Prototype frequencies in an image form a histogram
- Histograms used for categorization

## 2. Visual Word Ambiguity by Kernel Codebooks

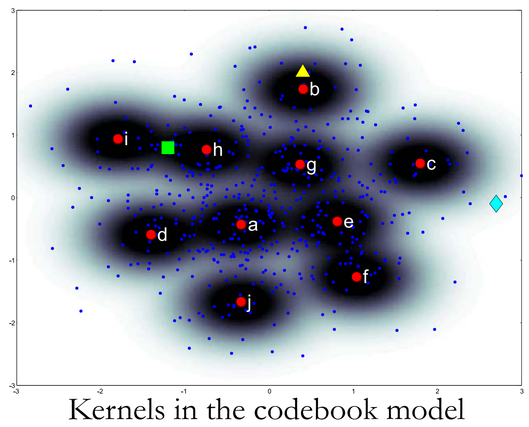



Two drawbacks with hard-assignment

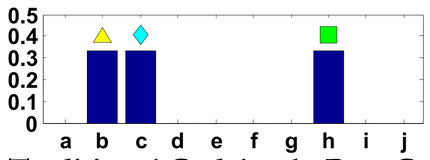
Gaussian Kernel $K_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2}\frac{x^2}{\sigma^2})$



Symmetric kernel on bins = Symmetric kernel on features

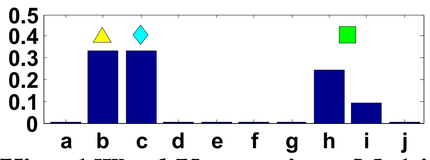



Kernels in the codebook model



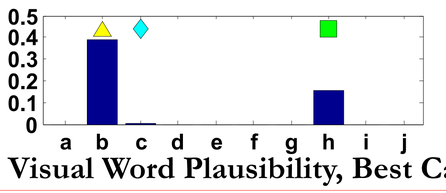

$$w = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} 1 & \text{if } w = \arg\min_{v \in V}(D(v, r_i)) \\ 0 & \text{otherwise} \end{cases}$$

**Traditional Codebook, Best Candidate, Constant Weight**



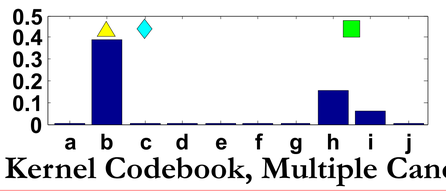

$$w = \frac{1}{n}\sum_{i=1}^{n} \frac{K_\sigma(D(w, r_i))}{\sum_{j=1}^{|V|} K_\sigma(D(v_j, r_i))}$$

**Visual Word Uncertainty, Multiple Candidates, Constant Weight**




$$w = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} K_\sigma(D(w, r_i)) & \text{if } w = \arg\min_{v \in V}(D(v, r_i)) \\ 0 & \text{otherwise} \end{cases}$$

**Visual Word Plausibility, Best Candidate, Kernel Weighted**

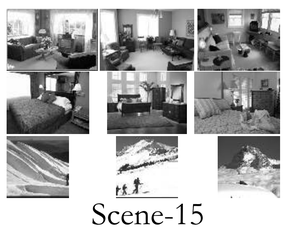



$$w = \frac{1}{n}\sum_{i=1}^{n} K_\sigma(D(w, r_i))$$

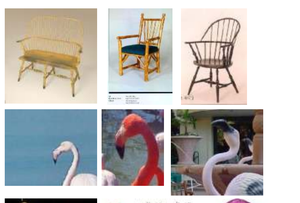
**Kernel Codebook, Multiple Candidates, Kernel Weighted**

- A drawback of the codebook model is that an image feature is assigned to the single best codeword
- Hard assignment yields 2 issues:
  1) Visual word Uncertainty (eg ■)
  2) Visual word Plausibility (eg ♦)
- We propose ambiguity modeling by kernel density estimation
- A symmetric kernel, allows placing the kernel on the visual words
- We separately model each of the four variations of kernel codebooks
- $w$ =Visual word, $V$ =Vocabulary, $D$ =Distance $n$ =number of features

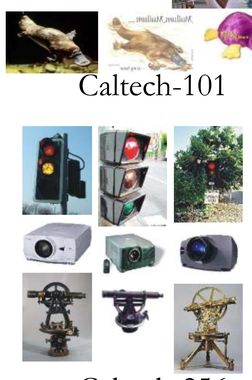
## 3. Experiment on 3 Data sets: Scene-15, Caltech-101, Caltech-256



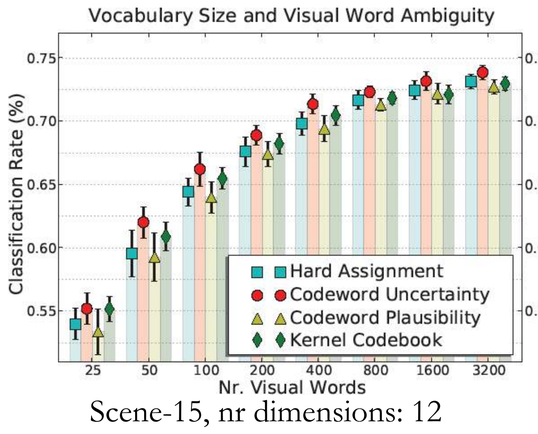
Scene-15



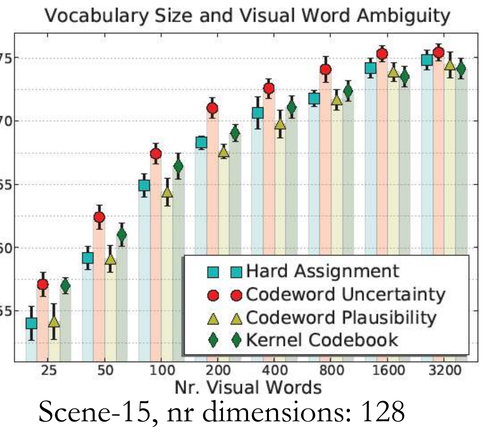
Caltech-101



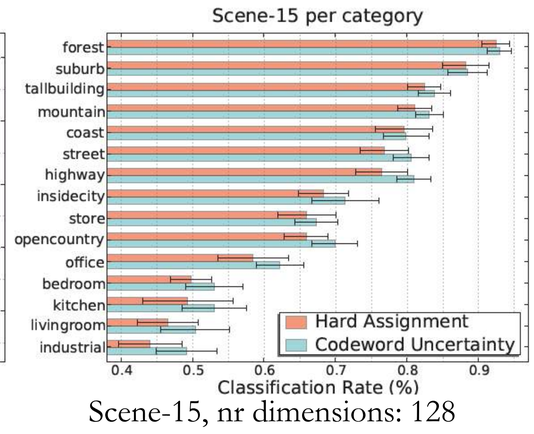
Caltech-256



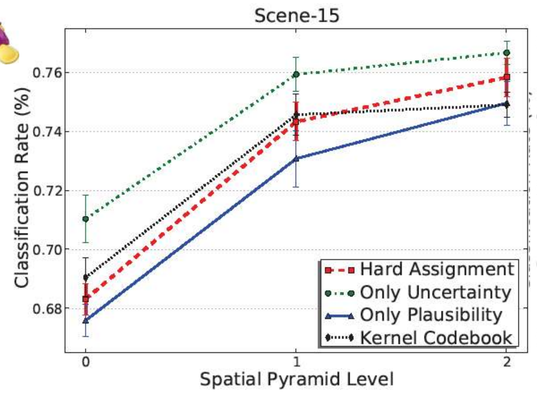

Scene-15, nr dimensions: 12

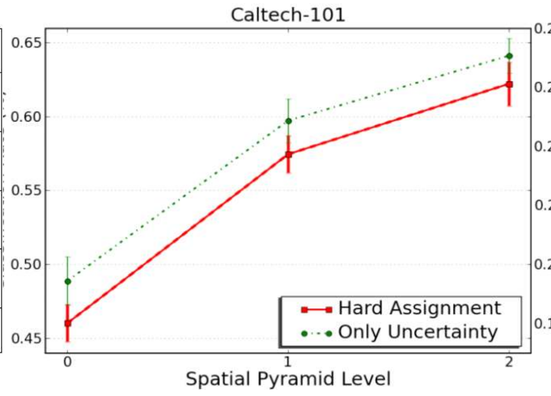


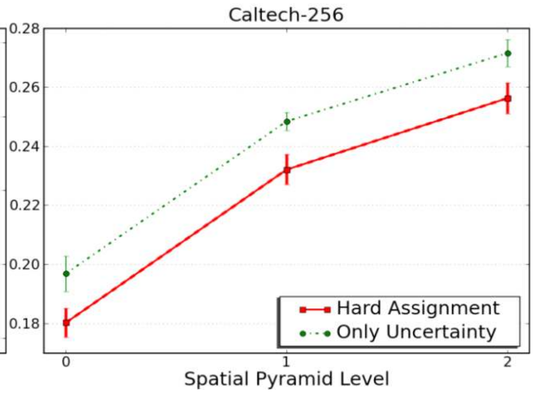

Scene-15, nr dimensions: 128




Scene-15, nr dimensions: 128




- Experimental questions:
  1) Evaluate 4 kernel codebook types
  2) Effect of feature dimensionality
  3) Influence of vocabulary size
  4) Test increasing nr of categories
- Scene-15: 4,485 images
- Caltech-101: 8,677 images
- Caltech-256: 29,780 images
- Sift descriptor (128 dim)
- Sift after PCA (12 dim)
- 8 vocabulary sizes: {25, 50, 100, 200, 400, 800, 1600, 3200}
- Test modularity by incorporating kernel codebooks in Lazebnik's Spatial Pyramid
- Repeat each experiment 10x

## 4. Conclusions

| Categories | Train set size | Test set size | Performance Increase |
|---|---|---|---|
| Scene-15 | 1500 | 2985 | 4.0 ± 1.7 % |
| Caltech-101 | 3030 | 5050 | 6.3 ± 1.9 % |
| Caltech-256 | 7680 | 6400 | 9.3 ± 3.0 % |

- Codeword Plausibility hurts performance
- Our approach is more robust to the curse of dimensionality than the traditional codebook model
- Larger vocabularies increase performance asymptotically, mostly benefitting hard assignment
- The relative performance gain of ambiguity modeling increases as the number of categories grows
- Codeword Uncertainty yields best results, over all dimensions, vocabulary sizes, and datasets

MultimediaN