Bag of Words

- Bag-of-words
- (Spatial) pyramid matching
- Matlab demo

Text Analysis

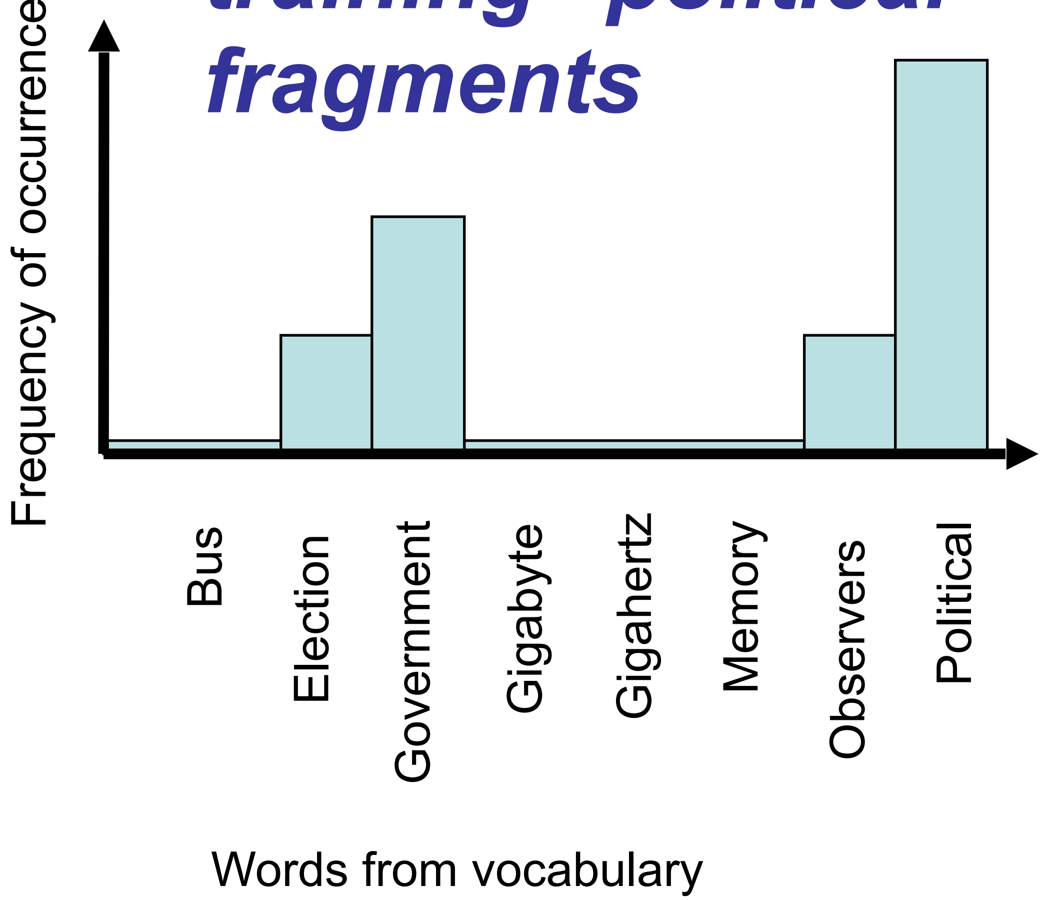
Political observers say that the government of Zorgia does not control the political situation. The government will not hold elections …

The ZH-20 unit is a 200Gigahertz processor with 2Gigabyte memory. Its strength is its bus and high-speed memory……

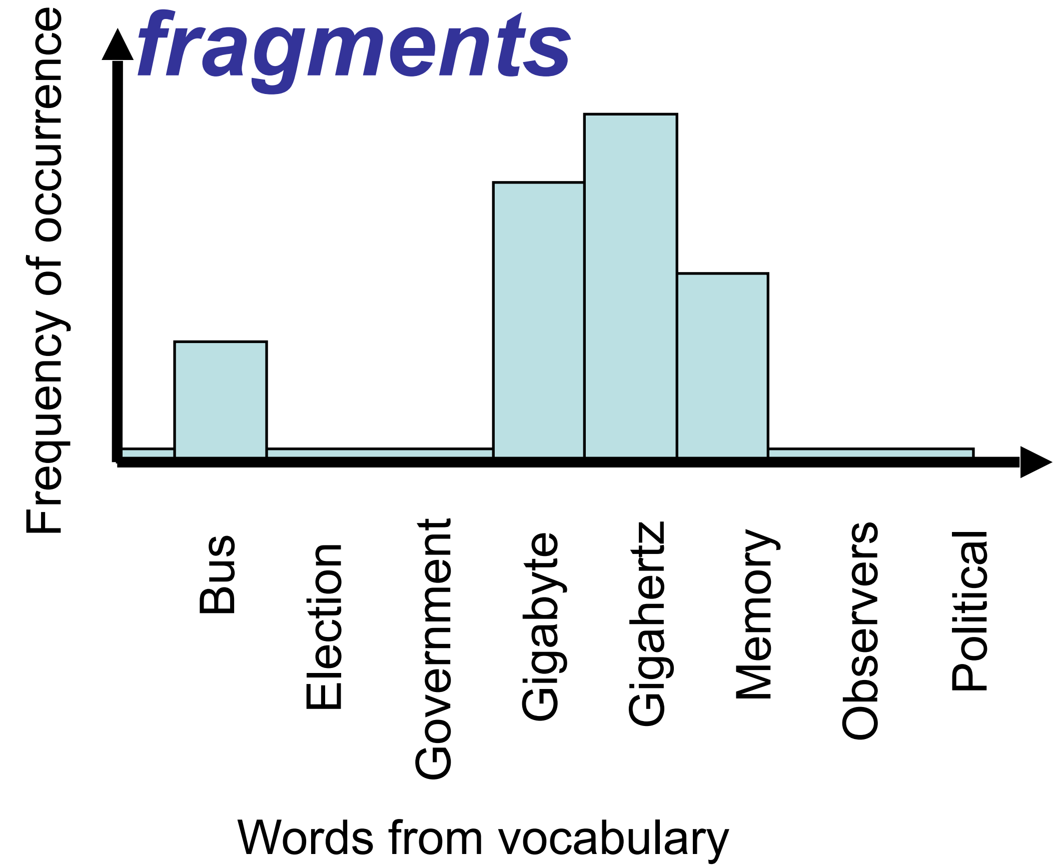
How to compare the two articles?

Bag-of-words

Histogram from training "political" fragments



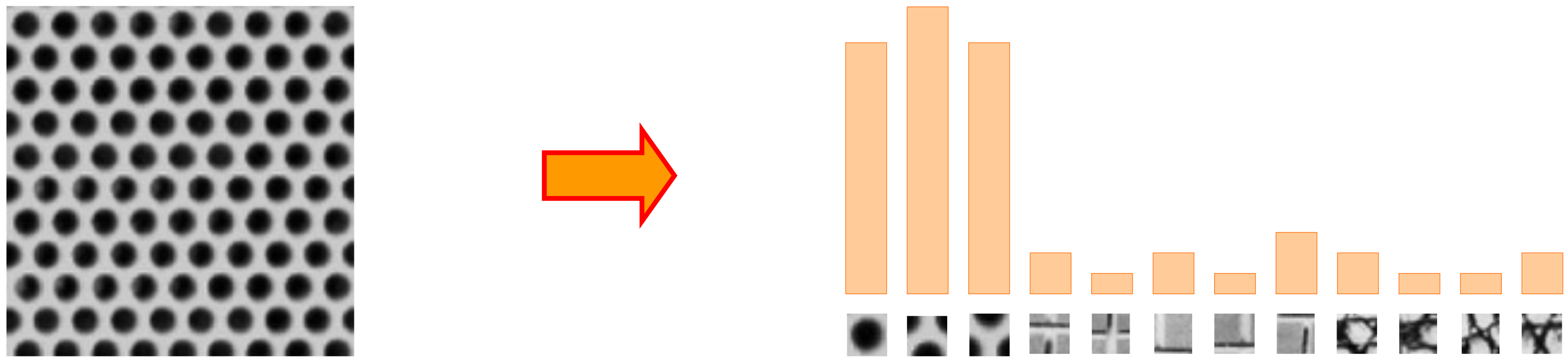
Histogram from training "computer" fragments

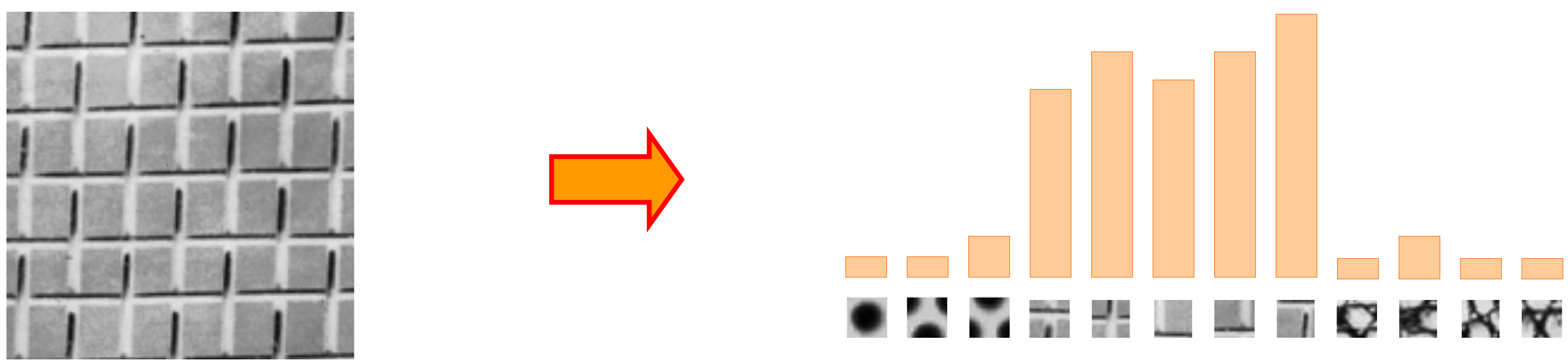


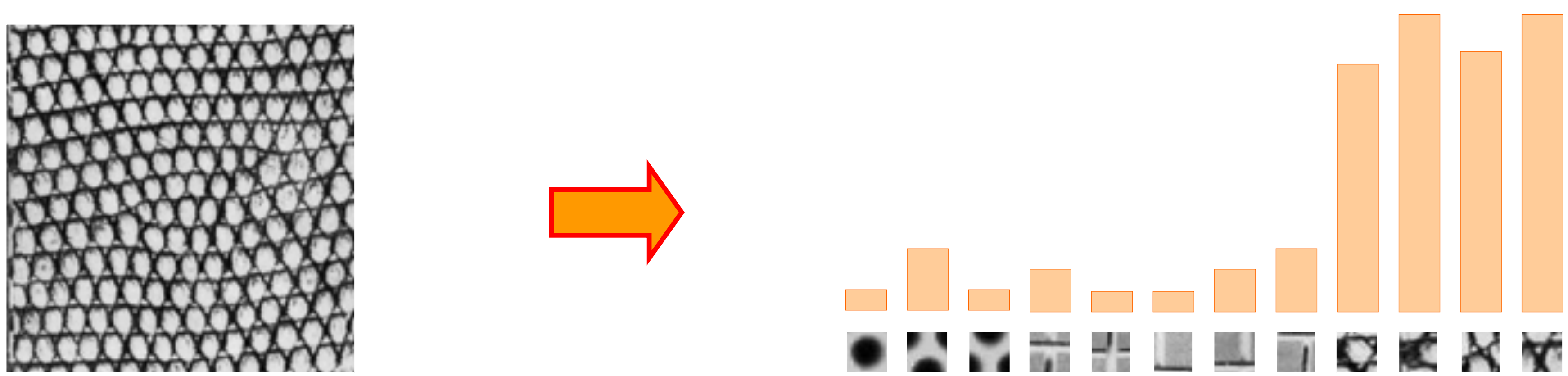
Compare histograms

Analogy:
Text fragment ←→ Image region
Word ←→ Texton

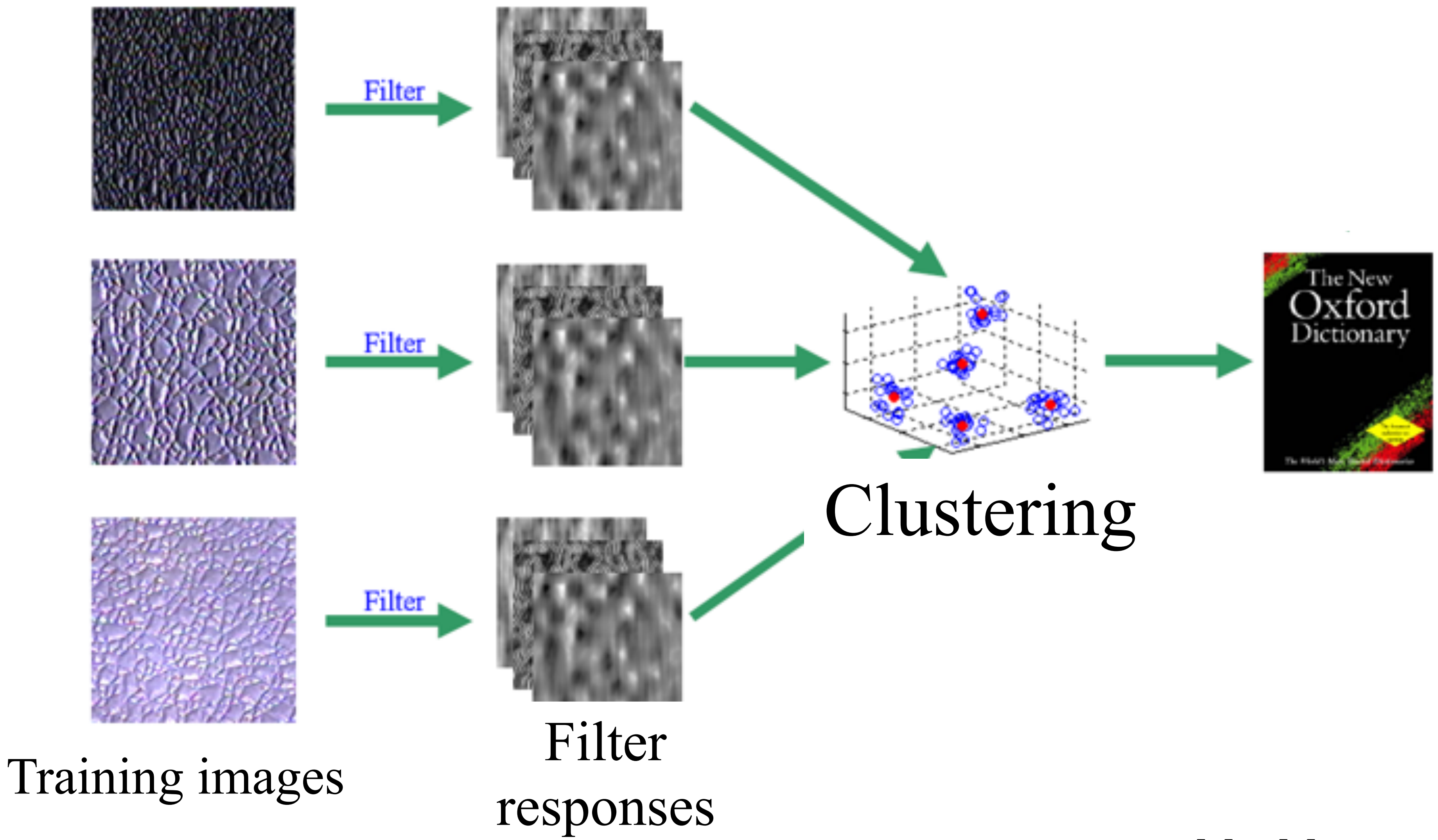
Representing textures as bags-of-visual words







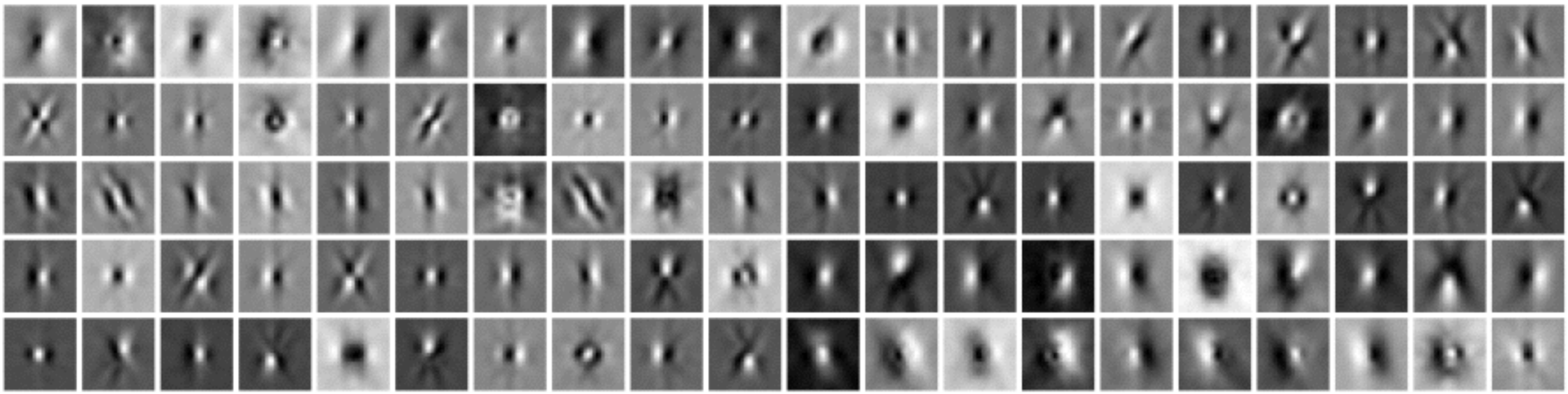
Bag-of-visual-words



Given a large set of vectorized image patches: $x \in R^{M \times M} \Rightarrow x \in R^{M^2}$
and a bank of vectorized filters F = $[f_1, f_2, .. f_b]$

1. Project each patch into *basis* spanned by F: $y = F^T x, \quad y \in R^b$
   (does this basis span R^{M^2}? Is it orthonormal?)
2. Cluster patches in this projected space

Use pseudoinverse of filter bank to visualize cluster means in original space



Given a M X M image patch ‘x’ (reshaped into a M^2 vector) and a filter bank of B filters, filter bank responses can be seen as a change of basis

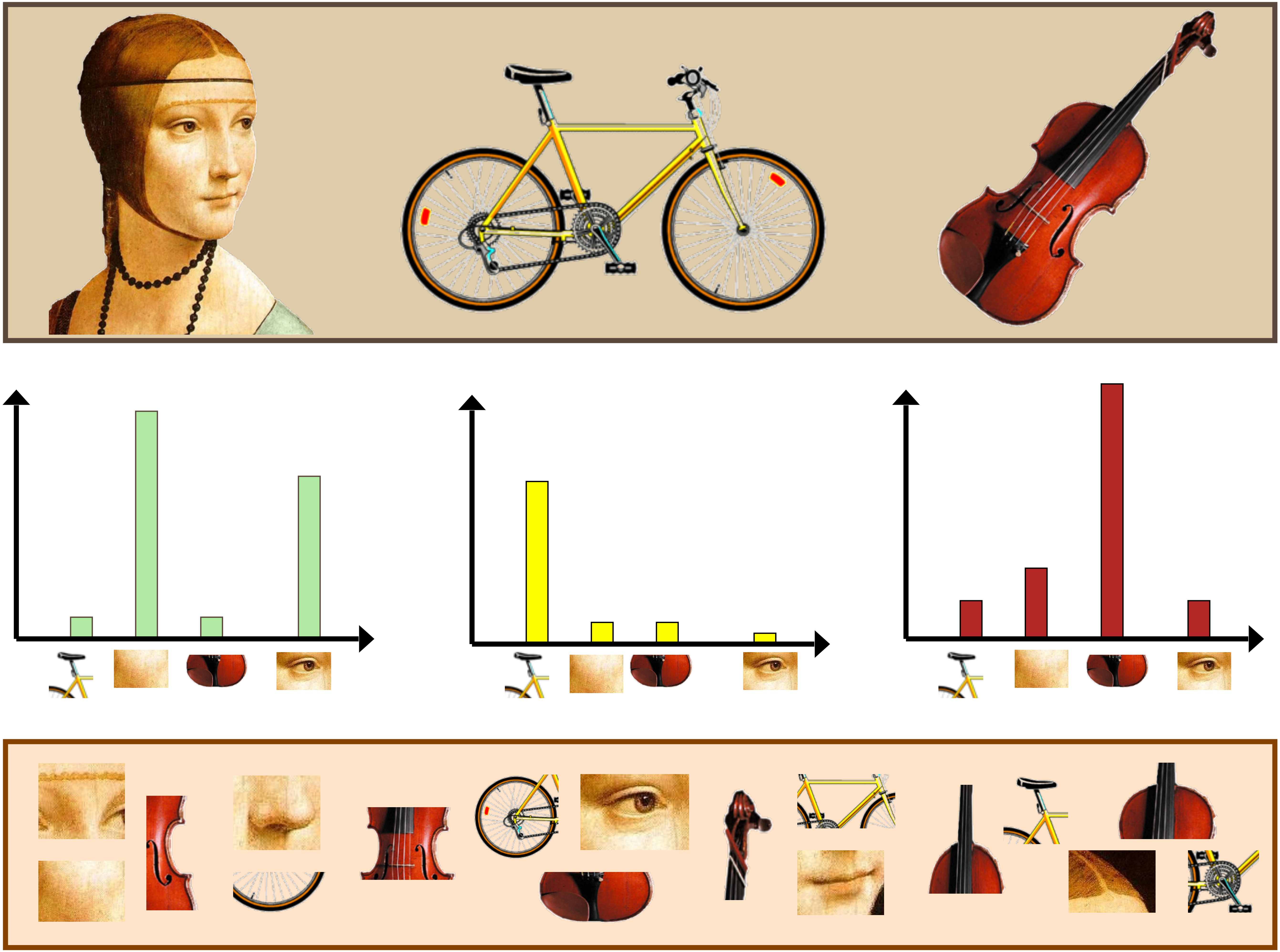
$$y = F^T x, \quad x \in R^{M^2}, y \in R^B$$

$$x \approx (F^T)^+ y$$

$$Vis(d_j) \approx (F^T)^+ d_j$$

Object → **Bag of ‘words’**

ICCV 2005 short course, L. Fei-Fei



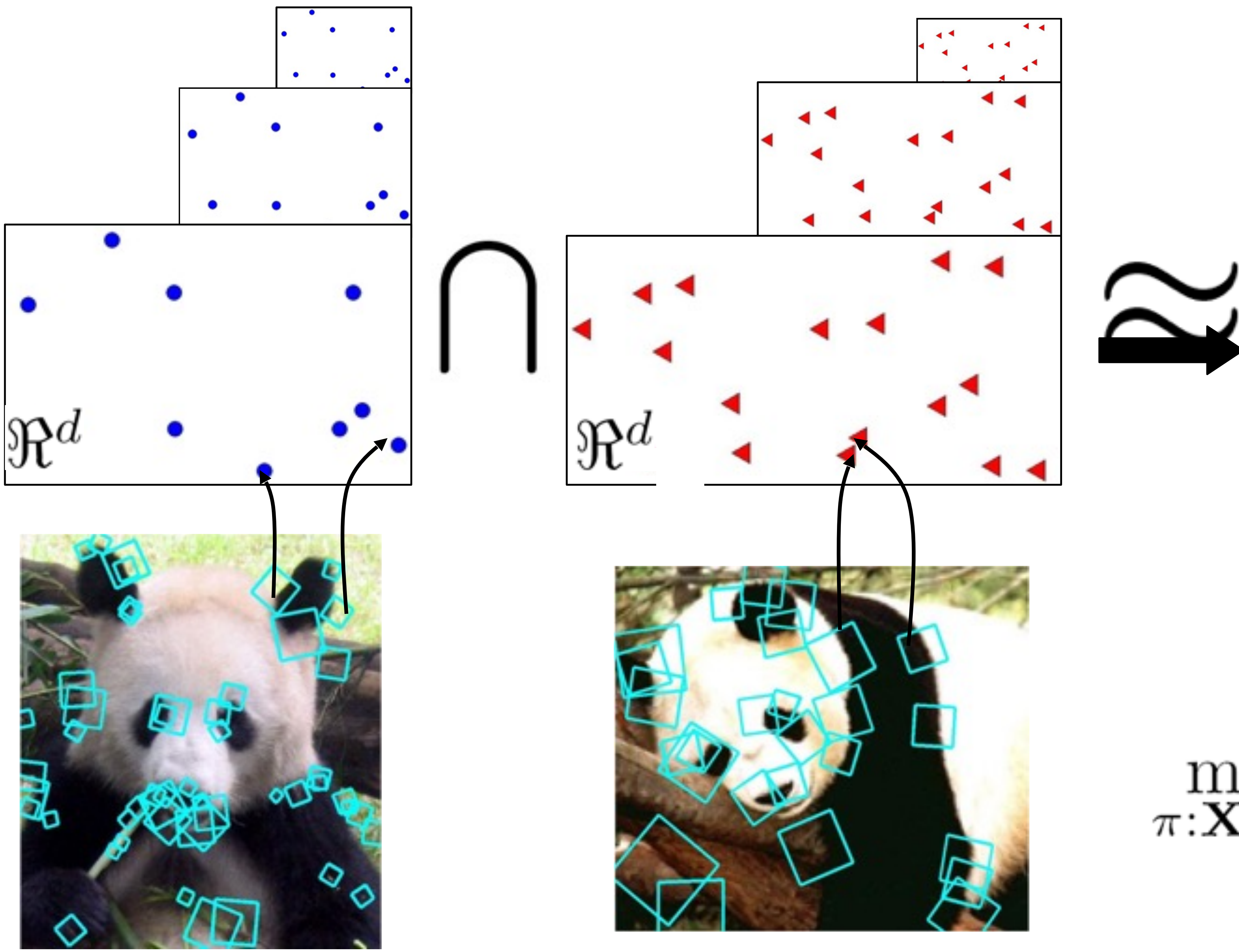
Recognition with bag-of-words

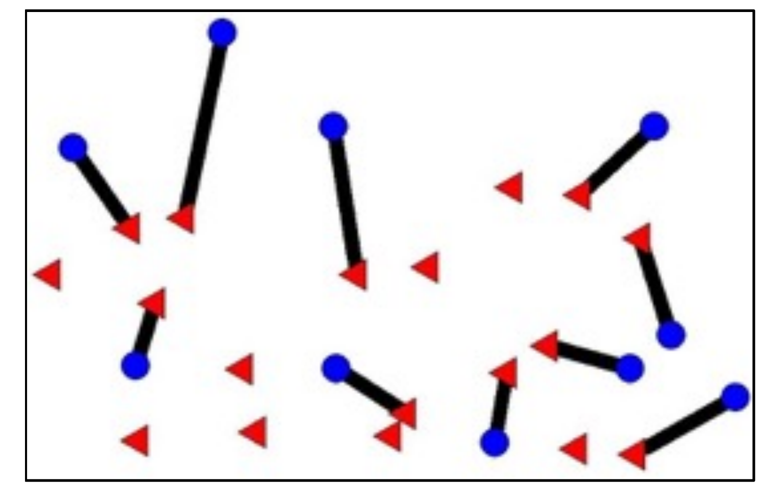
- Summarize entire image based on its distribution (histogram) of word occurrences.
- Compare to stored library of images (or class-specific *models*)

Image credit: Fei-Fei Li

Digression: alternative to quantization

Krauman & Darrell





optimal partial matching

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

$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$$
$$\vec{\mathbf{x}}_i \in \Re^d$$

$$\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$$
$$\vec{\mathbf{y}}_i \in \Re^d$$

Pyramid match kernel

Number of newly matched pairs at level *i*

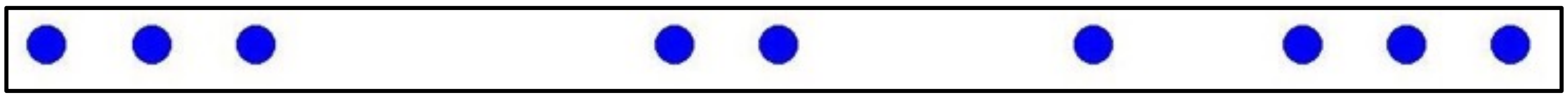
Approximate partial match similarity

$$K_\Delta = \sum_{i=0}^{L} w_i N_i$$

Measure of difficulty of a match at level *i*

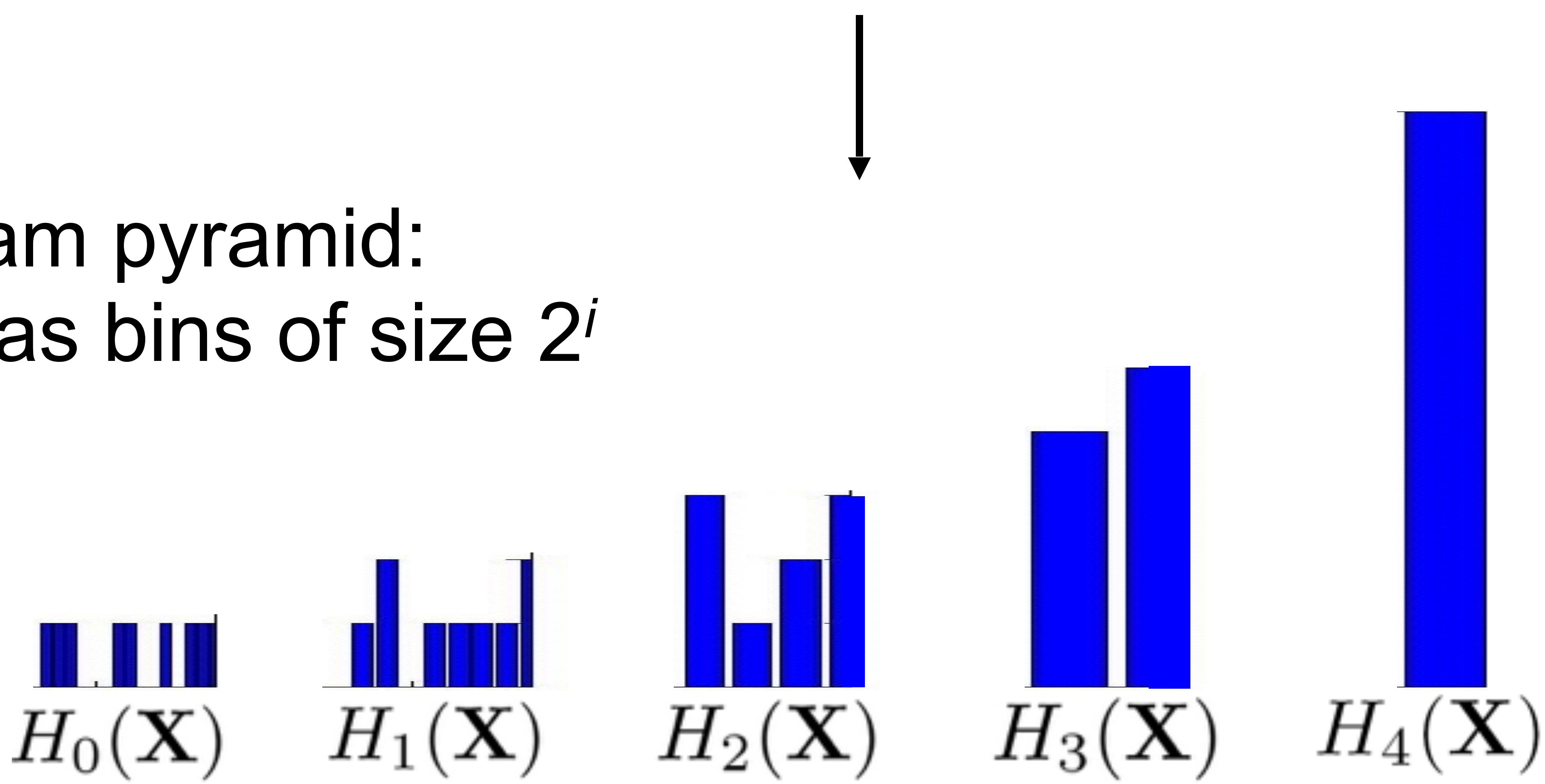
Feature extraction

$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}, \quad \vec{\mathbf{x}}_i \in \Re^d$$



$d = 1$

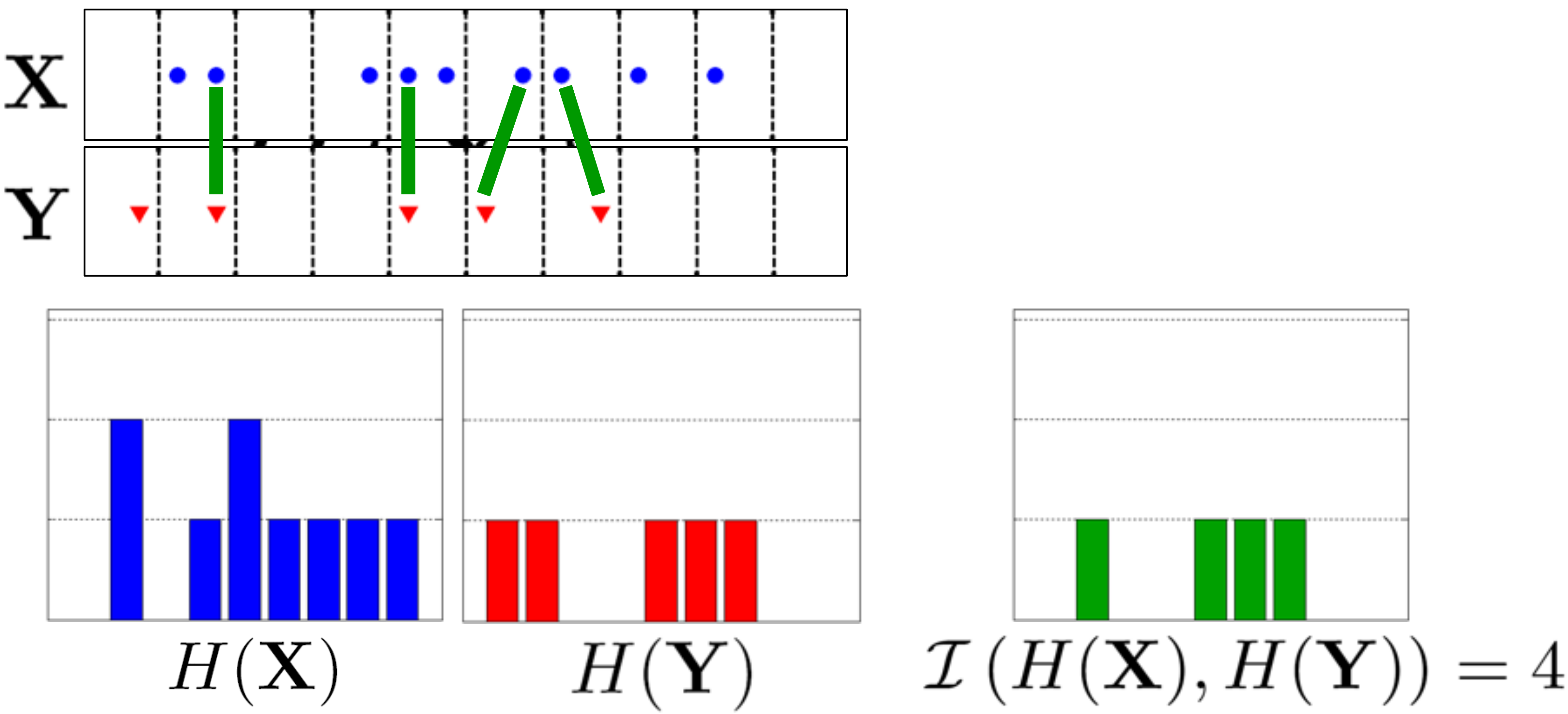
Histogram pyramid:
level *i* has bins of size 2^i



$$\Psi(\mathbf{X}) = [H_0(\mathbf{X}), \ldots, H_L(\mathbf{X})]$$

Counting matches

Histogram intersection
$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$



Counting new matches

Histogram intersection

$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$

$$N_i = \underbrace{\mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right)}_{\text{matches at this level}} - \underbrace{\mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)}_{\text{matches at previous level}}$$

Difference in histogram intersections across levels counts *number of new pairs* matched

Pyramid match kernel

$$K_\Delta\left(\overbrace{\Psi(\mathbf{X}),\Psi(\mathbf{Y})}^{\text{histogram pyramids}}\right) =$$

$$\sum_{i=0}^{L} \underset{\substack{\uparrow \\ \text{measure of difficulty of a} \\ \text{match at level } i}}{\frac{1}{2^i}} \left(\underbrace{\mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)}_{\text{number of newly matched pairs at level } i}\right)$$

- Weights inversely proportional to bin size
- Normalize kernel values to avoid favoring large sets

Spatial Pyramid Matching

Quantize features into words, but build pyramid in space

Multiresolution representations are powerful!

Original images

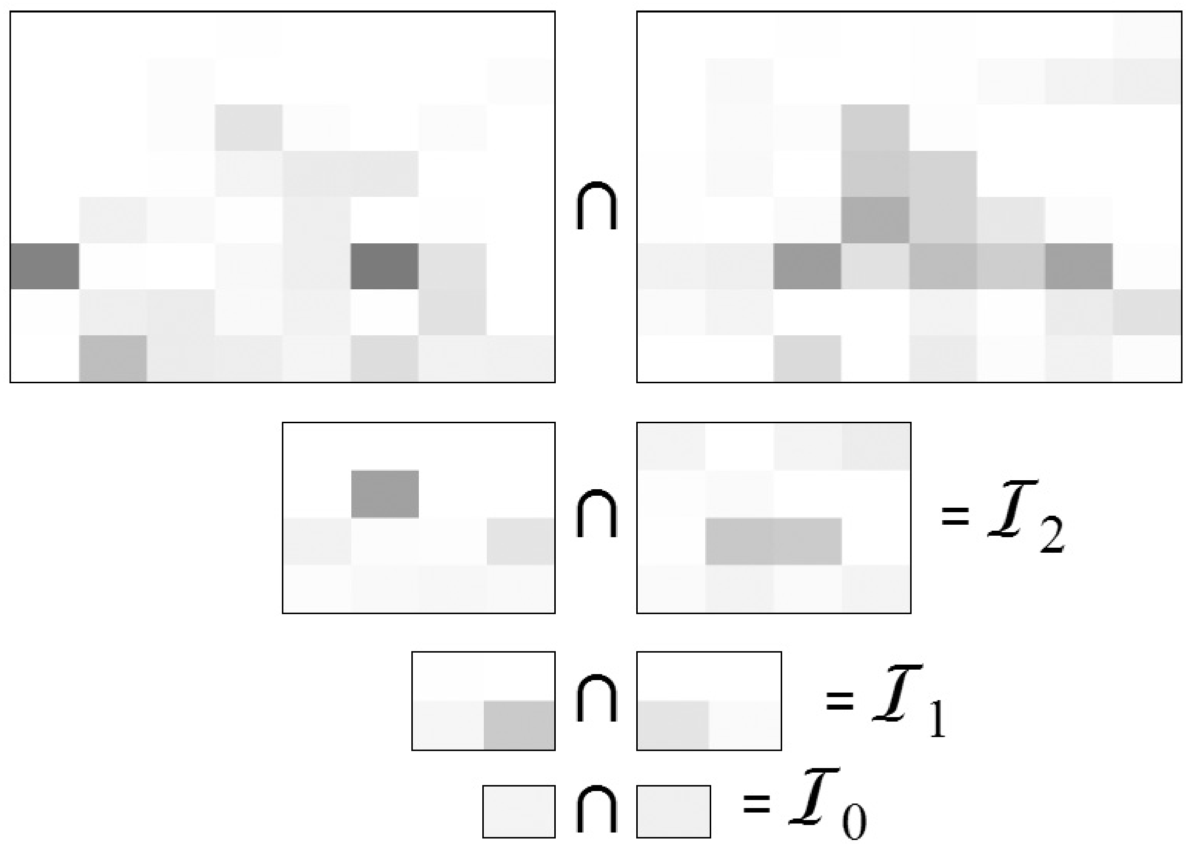
Feature histograms:

Level 3

Level 2

Level 1

Level 0



$\cap = \mathcal{I}_3$

$\cap = \mathcal{I}_2$

$\cap = \mathcal{I}_1$

$\cap = \mathcal{I}_0$

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

Outline

- Efficiency (pyramids, separability, steerability)
- Linear algebra
- Bag-of-words
- Frequency analysis (don't expect to get to)