Visual Object Recognition

state-of-the-art techniques and performance evaluation

Lecture 5: Metrics for Matching

- Descriptor distances
  - Chi^2 distance
  - Earth Mover's Distance (EMD)
- Ratio Score
- Visual Words
- Learning the metric

Descriptor Distances

- For a descriptor **q** in a query image. Which prototype in memory ($\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N$) is *most likely* to correspond to the same world object.

- Assuming additive i.i.d. Gaussian noise on all elements:

$$p(\mathbf{q}|\mathbf{p}_k) \propto \prod_{l=1}^{D} e^{-.5(p_{kl}-q_l)^2/\sigma^2}$$

$$\max(p) \quad \Leftrightarrow \quad \min(-\log(p))$$

$$-\log(p(\mathbf{q}|\mathbf{p}_k)) \propto \sum_{l=1}^{D} (p_{kl} - q_l)^2$$

Descriptor Distances

- So, the match with smallest distance is most likely correct, assuming i.i.d. Gaussian noise.
- What about the scalar product for normalised vectors/NCC?

$$||\mathbf{p} - \mathbf{q}||^2 = \mathbf{p}^T\mathbf{p} + \mathbf{q}^T\mathbf{q} - 2\mathbf{p}^T\mathbf{q} = 2(1 - \mathbf{p}^T\mathbf{q})$$

- But are all values identically distributed?
- ...are they independent?

Chi² Distance

- Many descriptors are histogram-like in their nature.
- For histograms, the histogram values typically follow the (discrete) *Poisson distribution*:

$$P(k|\mu) = \mu^k e^{-\mu}/k!$$

- With the statistics:

$$E[P(k)] = \mu \quad E[(P(k) - \mu)^2] = \mu$$

Chi² Distance

- For large values of μ, (e.g. 1000) a (continuous) Gaussian can approximate the Poisson distribution:

$$p(k|\mu) \approx \frac{1}{\mu\sqrt{2\pi}} e^{-.5(k-\mu)^2/\mu}$$

- Again, assuming independence, this leads to a negative log likelihood proportional to:

$$-\log(p(\mathbf{q}|\mathbf{p}_k)) \propto \sum_{l=1}^{D} (p_{kl} - q_l)^2/\mu_l$$

Chi² Distance

- If we estimate the variance by:

$$\mu_l \approx (p_{kl} + q_l)/2$$

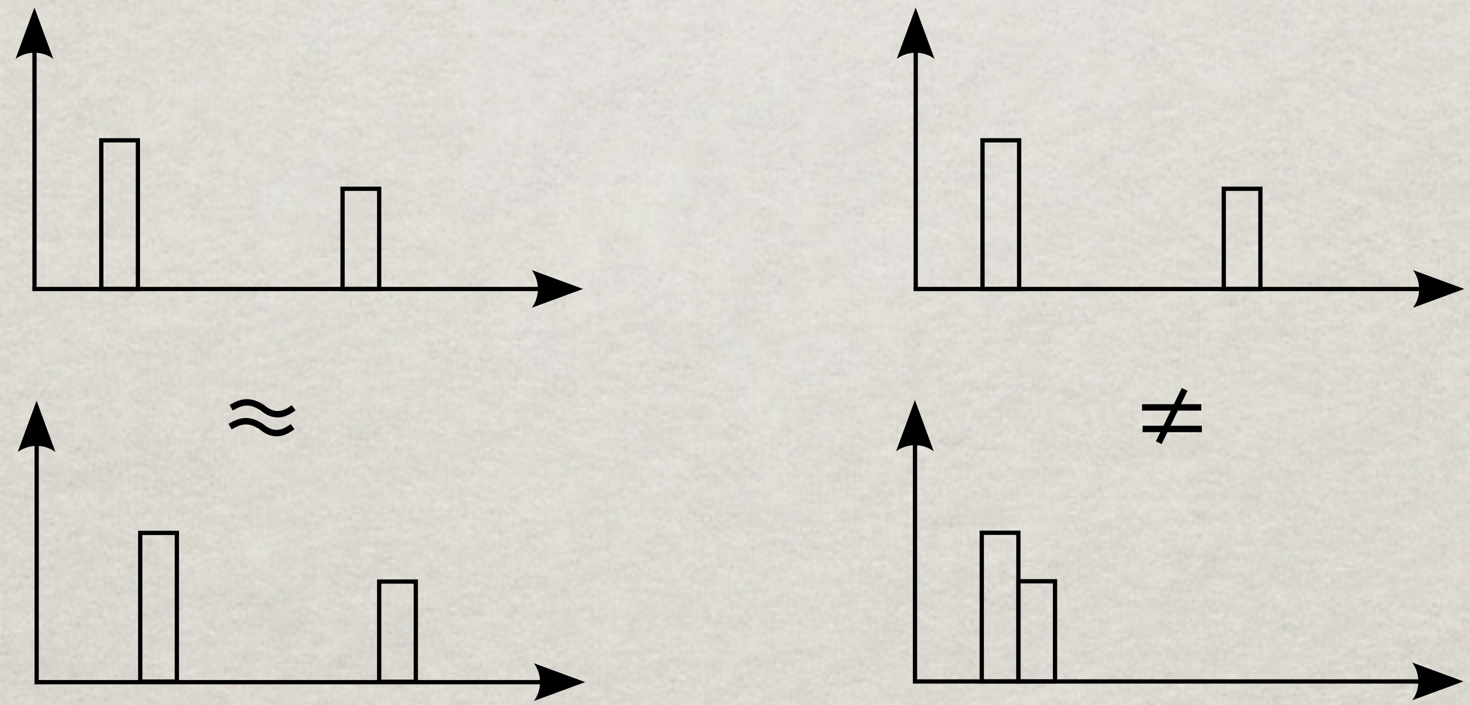
- We find that the most likely match is the one with the smallest Chi-squared distance:

$$\mathcal{X}^2(\mathbf{q}, \mathbf{p}_k) = \sum_{l=1}^{D} \frac{(p_{kl} - q_l)^2}{p_{kl} + q_l}$$

- This is assuming independence between bins.

Earth Mover's Distance

- In histograms, neighbouring bins are typically correlated



- Instead of falling in bin i, a sample is likely to fall in bin i+1.

Earth Mover's Distance

- Distance=cost of moving values in **p** to **q**
  cost=amount*distance

- First solve a linear programming problem:
  the transportation problem, Hitchcock 1941.

$$\min_{f_{ij}} \sum_{i=1}^{D} \sum_{j=1}^{D} f_{ij} d_{ij} \quad \text{here} \quad d_{ij} = |i-j|$$

- f_{ij} amount to move from i to j.

Earth Mover's Distance

- Transportation problem, cost function:

$$\min_{f_{ij}} \sum_{i=1}^{D} \sum_{j=1}^{D} f_{ij} d_{ij} \quad d_{ij} = |i - j|$$

- Constraints:

$$f_{ij} \geq 0 \quad \forall\, i, j \in [1, D]$$

$$\sum_{i=1}^{D} f_{ij} = q_j \quad \forall\, j \in [1, D]$$

$$\sum_{i=1}^{D} f_{ij} \leq p_j \quad \forall\, j \in [1, D]$$

Earth Mover's Distance

- Now compute EMD as:

$$d(\mathbf{p},\mathbf{q}) = \min_{f_{ij}} \frac{\sum_{i=1}^{D}\sum_{j=1}^{D} f_{ij}|i-j|}{\sum_{i=1}^{D}\sum_{j=1}^{D} f_{ij}}$$

- The denominator is needed if histograms are computed from different numbers of samples.
- Introduced in Computer Vision by Rubner&Tomasi&Guibas, at ICCV98
- Local expert: Thomas Kaijser

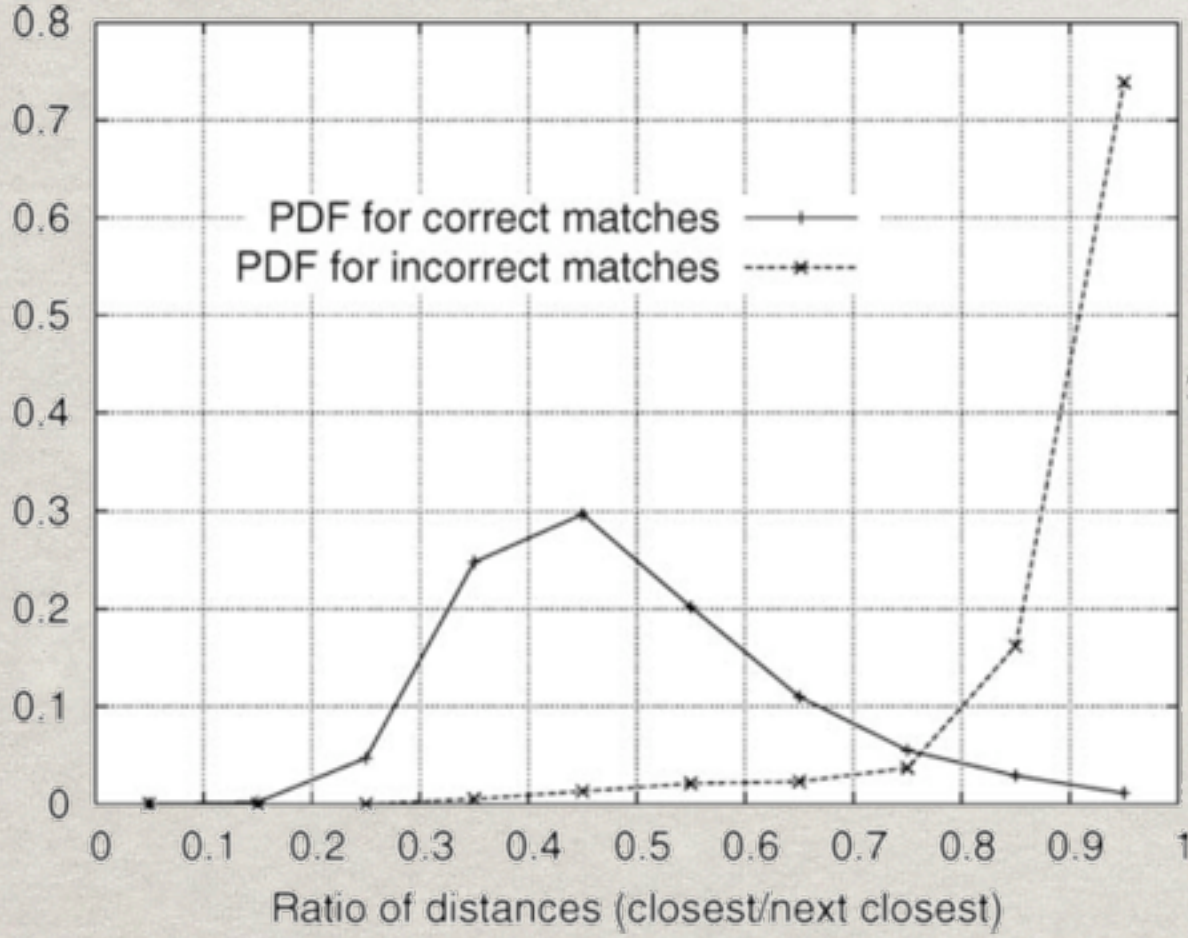
Ratio Score

- If we have best matches for descriptors $\mathbf{q}_1$ and $\mathbf{q}_2$ in the image. Which one is better?

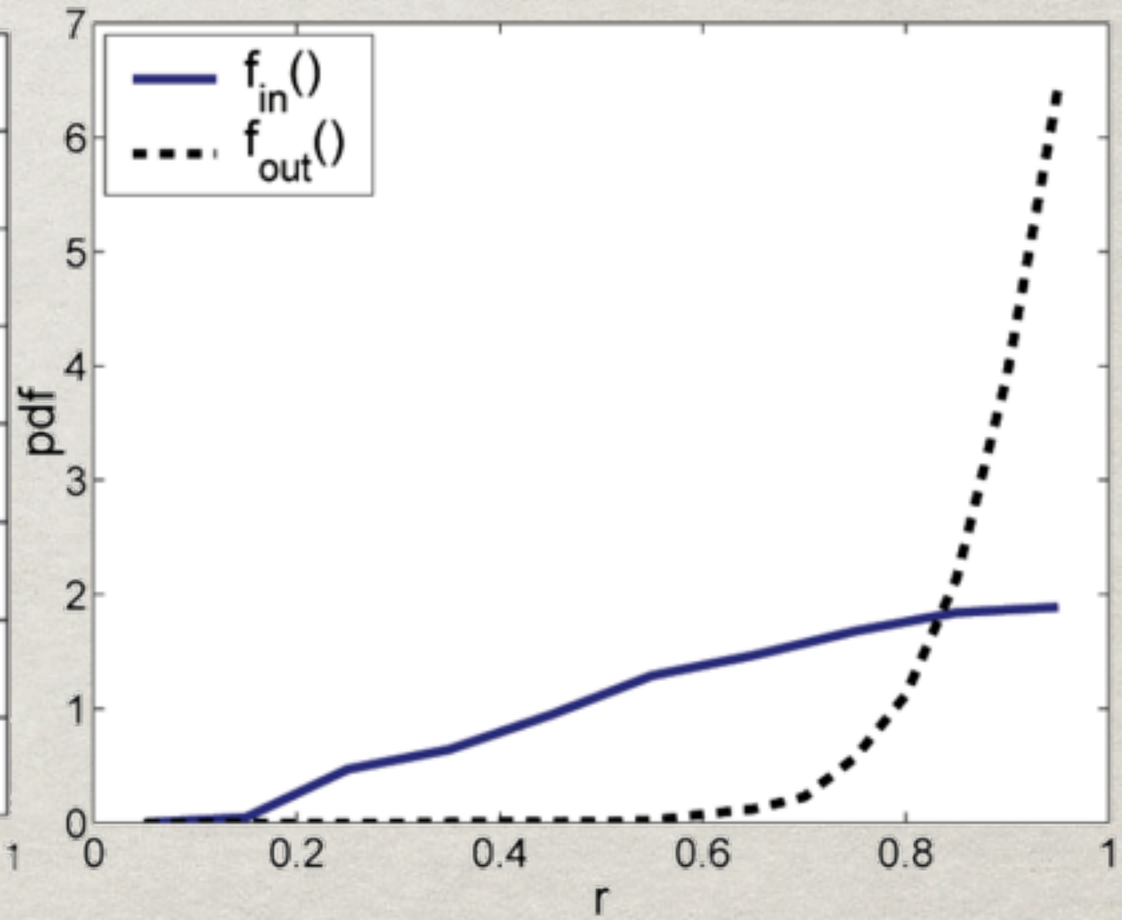
- Some features are *more common than others*, and by scoring the match for $\mathbf{q}_1$, according to the ratio between the best, and the second best match we can compensate for this:

$$r = d_1/d_2$$

Ratio Score



Lowe IJCV04



Goshen&Shimshoni PAMI08

Bags-of-Features and Visual Words

- A common technique for matching in large datasets. Sivic&Zisserman “Video Google” ICCV03.

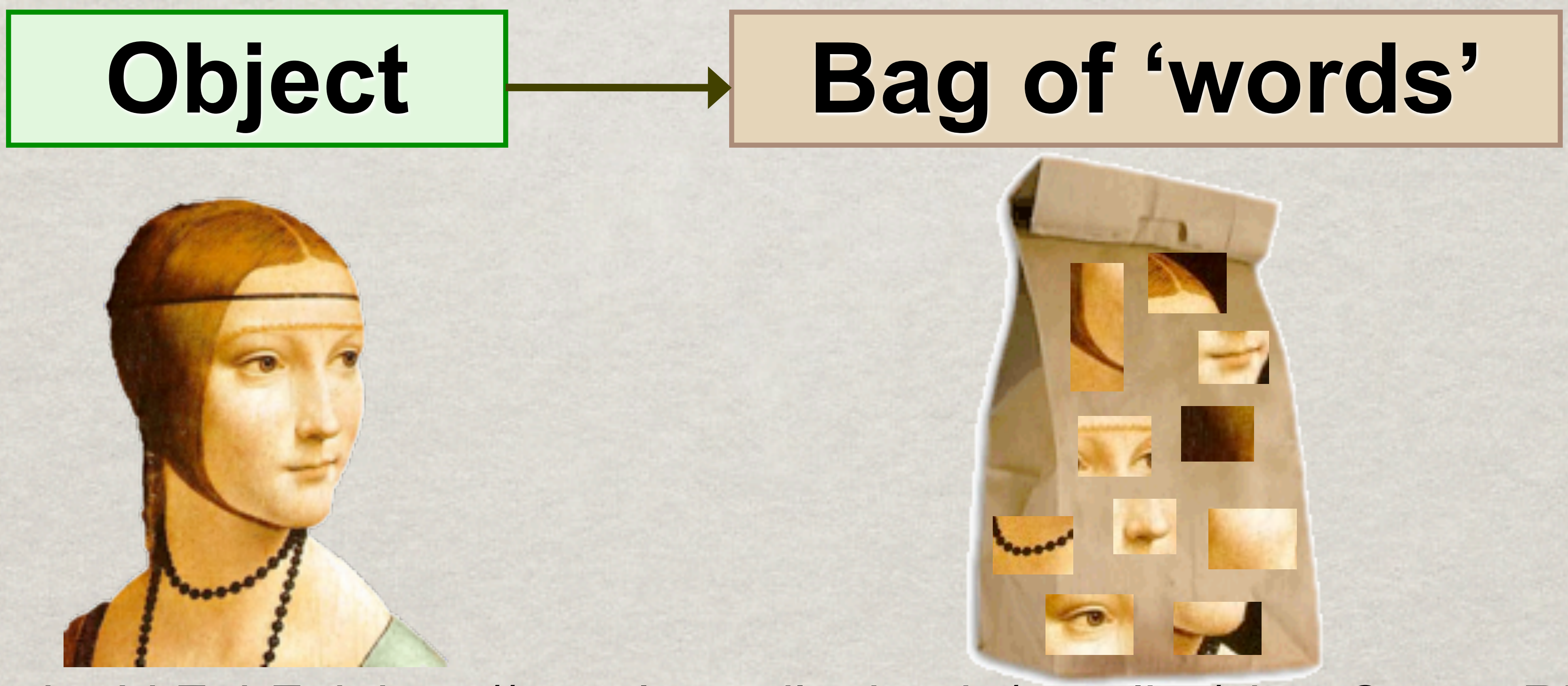


Illustration by Li Fei-Fei, http://people.csail.mit.edu/torralba/shortCourseRLOC/

- Completely disregards spatial relationships among features.

Visual Words

- Vector quantize feature space to into K parts using e.g. K-means clustering (e.g. function **kmeans** in Matlab) on large training set.

- Clustering is done in whitened space:

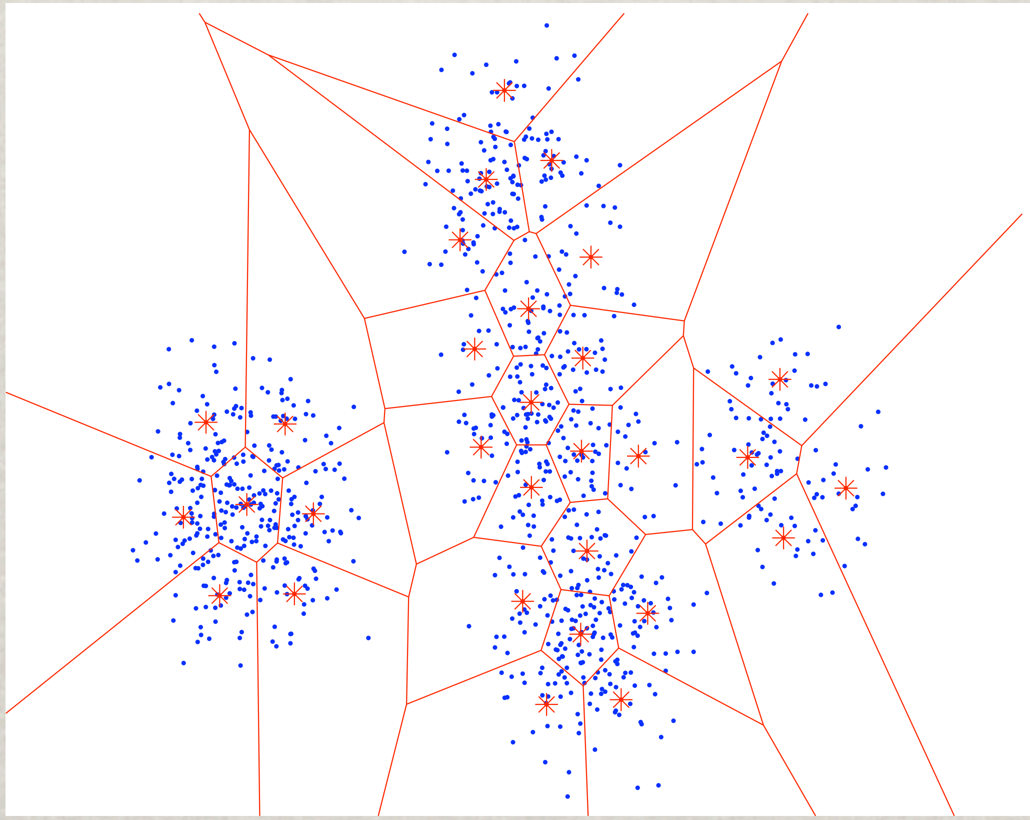
$$\hat{\mathbf{x}} = \mathbf{C}^{-1/2}(\mathbf{x} - \mu)$$

- Each descriptor is then approximated by the most similar prototype/visual word.

Visual Words

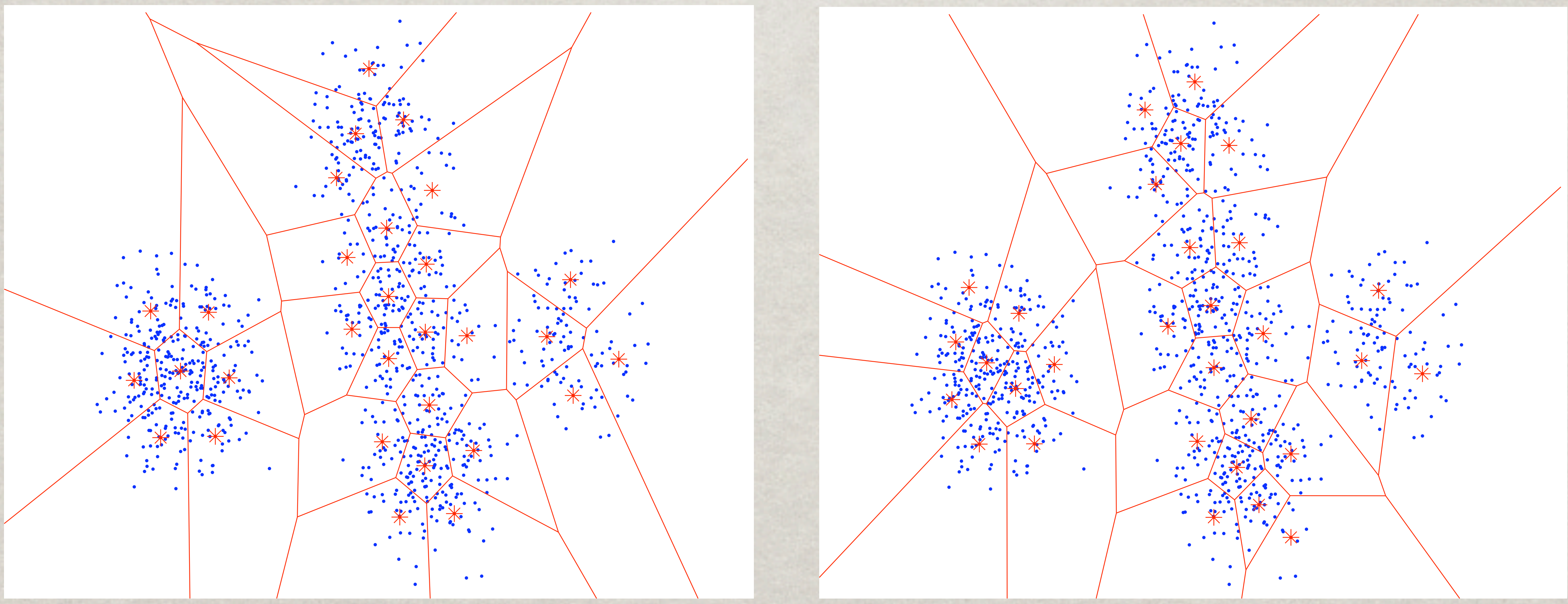
- K-means finds a local min of the following objective function (**x**-samples, **p**-prototypes):

$$J = \sum_{n=1}^{N} \min_{k \in [1 \ldots K]} ||\mathbf{x}_n - \mathbf{p}_k||^2$$



Visual Words

- Probability of visual words is somewhat equalized. cf. ratio score and histogram eq.
- K-means is not perfectly repeatable. (Try several times and pick highest J.)



Visual Words

- Analogy with text document matching.
- Each document (i.e. image) is represented as a vector of (TF-IDF) word frequencies

$$\mathbf{v}_d = \begin{pmatrix} v_1 & \dots & v_K \end{pmatrix}^T \qquad v_k = \frac{N_{kd}}{N_d} \log \frac{N}{N_k}$$

- term frequency: N_{kd}/N_d (word k, document d) Nistér&Stewénius CVPR06: skip N_d.
- inverse document frequency: N/N_k - inverse frequency of word k in whole database.

Visual Words

- Image matching is done by a normalised scalar product:

$$\hat{\mathbf{v}}_q^T \hat{\mathbf{v}}_p = \cos \phi$$

- An *inverted file* makes real-time matching possible on very large datasets:

  word1: frame 3, frame 17, frame 243...
  word2: frame 2, frame 23, frame 33...
  ...

Bag-of-Features

- Instead of the TF-IDF vector in the visual words method, one could simply compute a histogram of visual word occurrences.
- This is called a *bag-of-features*, or *bag-of-keypoints*
- With IDF as metric this is equivalent to the TF-IDF vector (when TF=N_{kd}).

Bag-of-Features

- The bag-of-features vector is often fed into a machine learning algorithm. (E.g. in today's article).

- Typically K is large and most values are zero.

  Csurka et al. K=1000
  Sivic&Zisserman K=6000 and 10,000
  Nistér&Stewénius K=16e6

Learning the Metric

- What we ultimately want is to distinguish good feature matches from bad.
- Collect known corresponding descriptors:

$$\{(\mathbf{p}_k, \mathbf{q}_k)\}_1^K \quad \text{and set} \quad \mathbf{d}_k = \mathbf{p}_k - \mathbf{q}_k$$

- We now want to find a linear transformation that makes the noise equal in magnitude in all directions:

$$\mathbf{y}_k = \mathbf{T}\mathbf{p}_k \quad \text{assuming} \quad \mathbf{d}_k \sim \mathcal{N}(0, \mathbf{C})$$

Learning the Metric

- Find a whitening transform $\mathbf{T}$ from the covariance matrix:

$$\mathbf{C} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{d}_k\mathbf{d}_k^T \quad \text{with} \quad \mathbf{T}\mathbf{C}\mathbf{T}^T = \mathbf{I}$$

- Valid solutions:

$$\mathbf{T} = \mathbf{R}\mathbf{C}^{-1/2} \quad \text{where} \quad \mathbf{R}\mathbf{R}^T = \mathbf{I}$$

- If we only use the first few dimensions we should choose **R** such that it selects dimensions where we "see things happen".

Learning the Metric

- Find **R** from PCA of transformed SIFT feature space:

$$\mathbf{C}_b = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n\mathbf{y}_n^T - \mathbf{m}\mathbf{m}^T \quad \mathbf{m} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n$$

$$\mathbf{R}\mathbf{D}\mathbf{R}^T = \mathbf{C}$$

- Final contraction operator:

$$\mathbf{P} = \bar{\mathbf{I}}_k\mathbf{R}\mathbf{C}^{-1/2}$$

- Where $\mathbf{I_k}$ is a k*128 truncated identity matrix.

Learning the Metric

- This Mahalanobis metric for features was published at ICCV07 by Mikolajczyk&Matas, SIFT 128->40 dim
- A similar method that only finds a rotation called linear discriminant embedding(LDE) also at ICCV07 by Hua&Brown&Winder, SIFT128->14/18dim
- Besides reducing dimensionality, these techniques also improve matching results.

Learning the Metric

- Linear Discriminant Embedding(LDE)
- Maximise

$$\mathbf{J}(\mathbf{w}) = \frac{\sum_{\text{outlier}(i,j)} \mathbf{w}^T(\mathbf{p}_i - \mathbf{q}_j)^2}{\sum_{\text{inlier}(i,j)} \mathbf{w}^T(\mathbf{p}_i - \mathbf{q}_j)^2}$$

$$\mathbf{J}(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{A}\mathbf{w}}{\mathbf{w}^T\mathbf{B}\mathbf{w}}\,, \quad ||\mathbf{w}|| = 1$$

- Where **A** covariance for outliers and **B** inliers.

Learning the Metric

- J($\mathbf{w}$) is maximised by eigenvectors with large eigenvalues in $\mathbf{B}^{-1}\mathbf{A}$
- Eigenvalues of $\mathbf{B}$ are set to $\tilde{\lambda}_i = \max(\lambda_i, \lambda_r)$

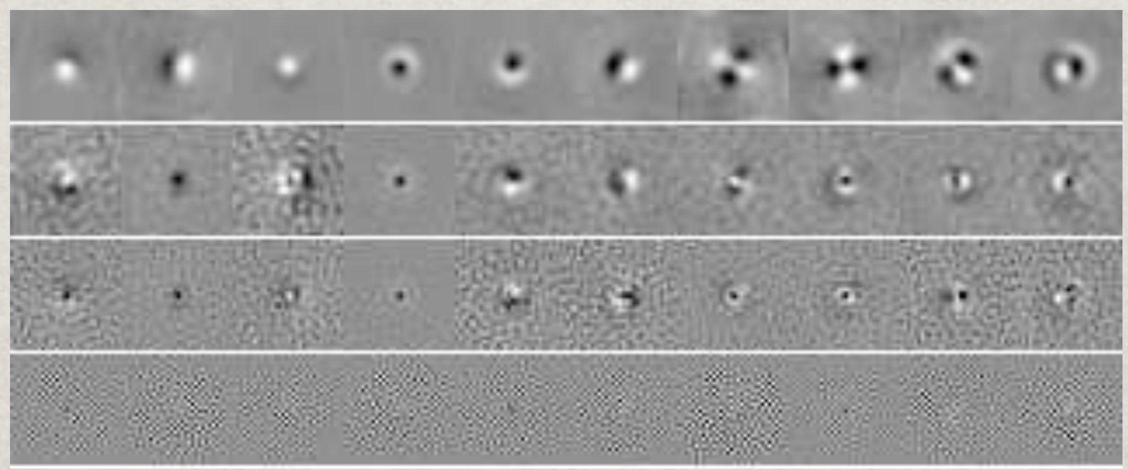
$$r = \arg\min_n \frac{\sum_{i=n}^{N} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \geq \alpha$$

- α can be interpreted as a threshold on SNR. This is called *Power Regularisation*
- Many variations of the algorithm in the paper.

Learning the Metric

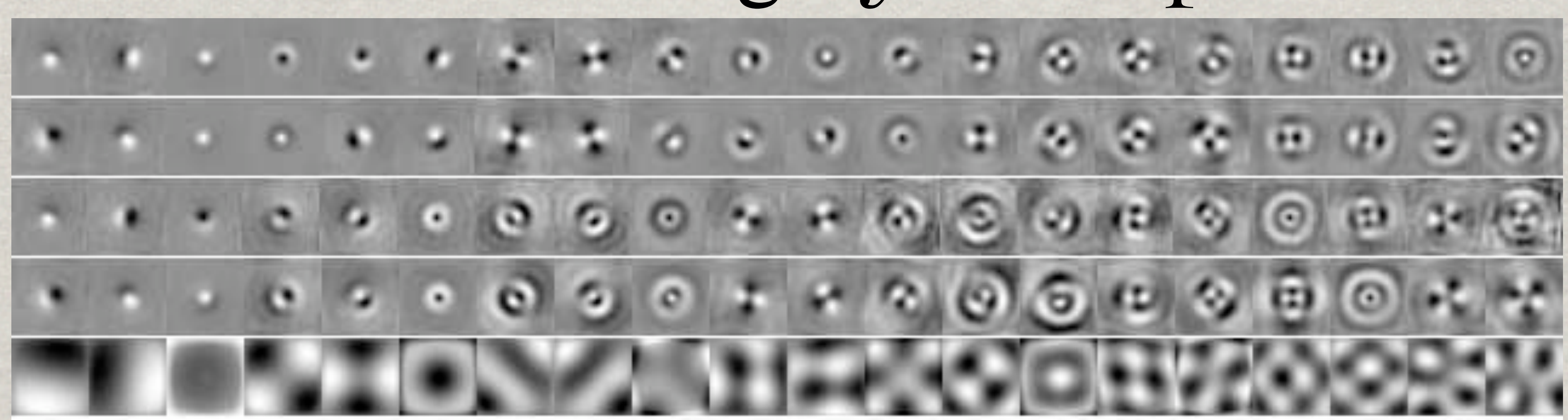
- Some LDE results on grey-scale patches:
- Reducing the amount of power reg:

20%
10%
2%
0%



- Linear filters found on grey-scale patches:

LDE-I
LDE-II
OLDE-I
OLDE-II
PCA



Discussion

- Questions/comments on paper and lecture.