Distances and Kernels

Amirshahed Mehrtash

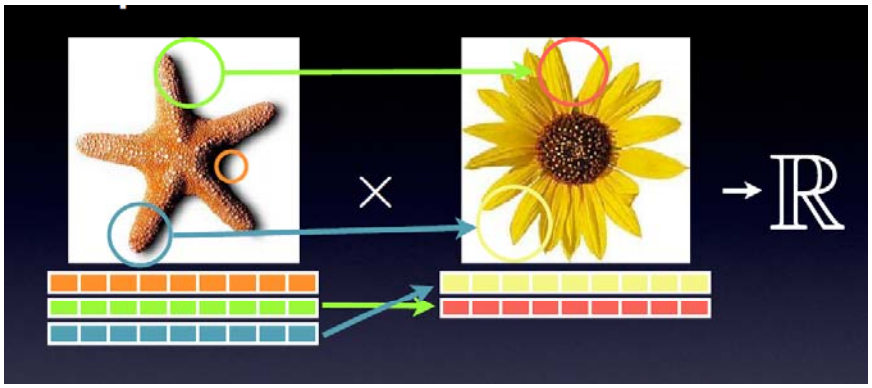
Motivation

How similar?

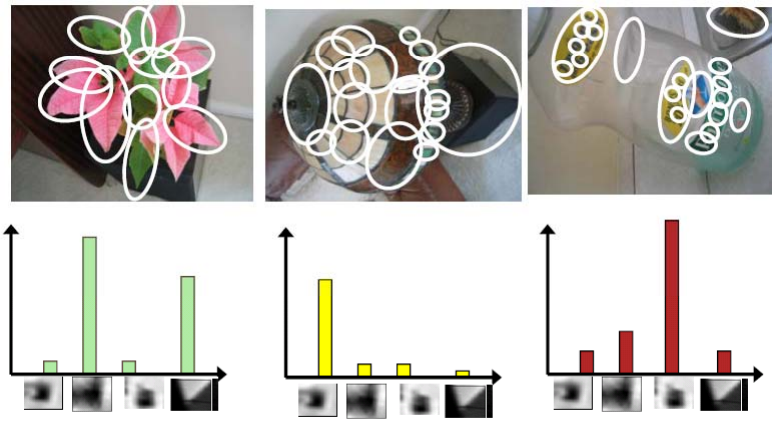
Problem Definition

- Designing a fast system to measure the similarity of two images.
- Used to categorize images based on appearance.
- Used to search for an image (part of an image) e.g. in a video.
- Used for object recognition

Patch based



Features



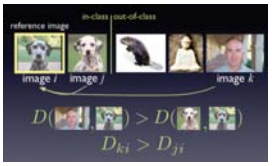
Outline

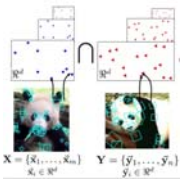
A. Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification, by A. Frome, Y. Singer, F. Sha, J. Malik. ICCV 2007.

B. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features, by K. Grauman and T. Darrell. ICCV 2005.

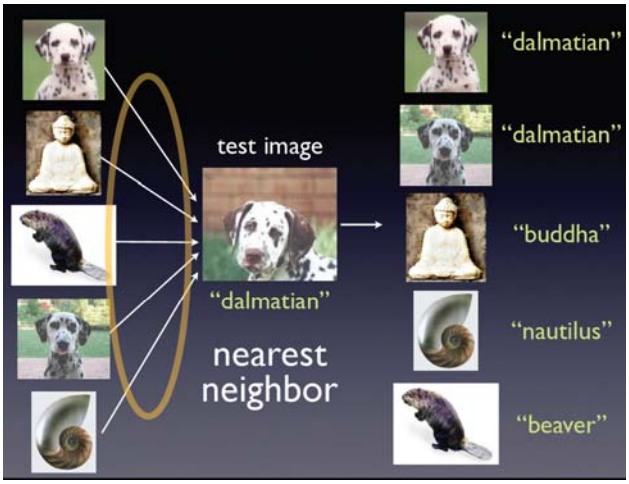
C. Video Google: A Text Retrieval Approach to Object Matching in Videos, J. Sivic and A. Zisserman, 2003.

D. Comparison and relevance.





Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification, by A. Frome, Y. Singer, F. Sha, J. Malik. ICCV 2007.



Andrea Frome's ICCV 2007 presentation

Distance function

- *A metric (distance function) d on a set X is a function such that:*

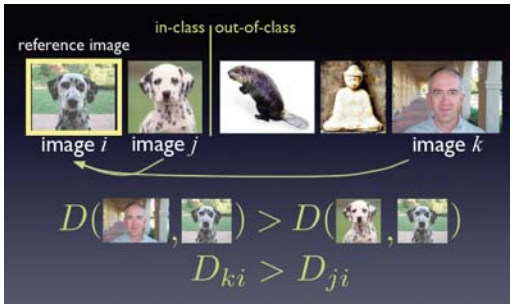
$d : X \times X \rightarrow \mathbf{R}$

1. $d(x, y) \geq 0$ (*non-negativity*)
2. $d(x, y) = 0$ if and only if $x = y$ (*identity of indiscernibles*)
3. $d(x, y) = d(y, x)$ (*symmetry*)
4. $d(x, z) \leq d(x, y) + d(y, z)$ (*subadditivity* / *triangle inequality*)

- Conditions 1 and 2 imply positive definiteness.
- Not independent ; 1 can be concluded from the others.

"Distance function"

- However we do not need a such a metric.
- Symmetry does not need to hold. Just as long as it gives lower values for objects in the same category versus two objects from different ones.

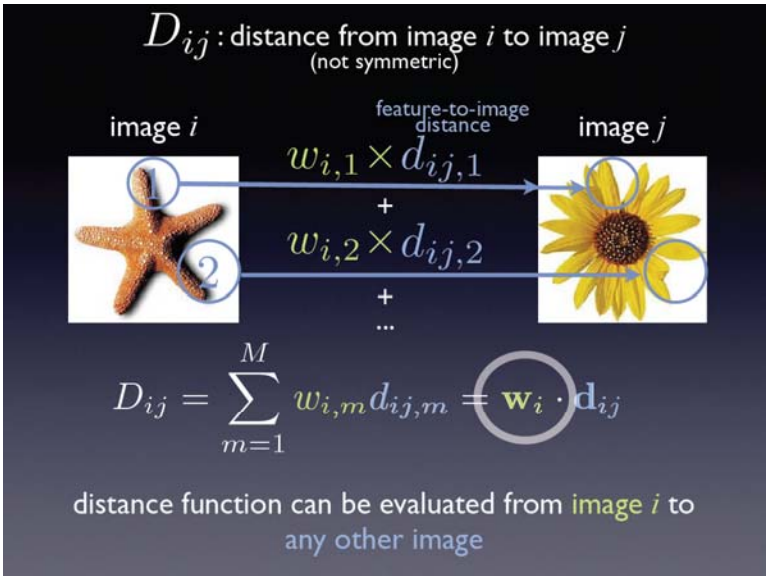


How to compute this "distance"

- This is a patch based approach (e.g. SIFT or geometric blur) and is done in three steps:

1. First find the distance between patch based shape feature descriptors in the two images.(each feature is a fixed length vector and the distance function here could be a simple L_1 or L_2 norm).
2. For every patch feature (m^{th}) from image i find the best matching (nearest neighbor) patch feature in image j ($d_{ij,m}$)
3. Define the image to image distance as a weighted sum of these patch to patch distances.

$$D_{ij} = \sum_{m=1}^{M} w_{i,m} d_{ij,m}$$

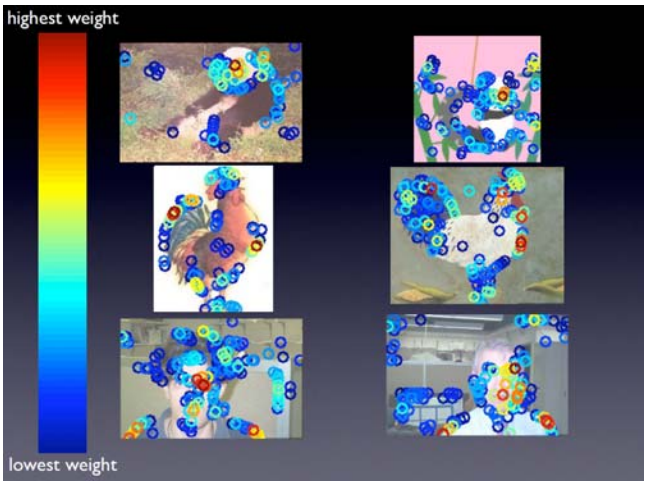


Andrea Frome's ICCV 2007 presentation

A note on the weights

- The weights basically signify the importance of a feature in each image (based on the category the image is in).
- For that very reason we can be robust to clutter\background as the weights assigned to their features are low.
- These weights are computed for any image we compare another image to.
- Once we have w_i we can compute the distance from image i to any other image.
- That is why the distance function is not symmetric since when we compare image i to image j we use w_j and when we compare image j to i we use w_i.
- The main problem here is to optimize these weights for every image.

An example of weights



Andrea Frome's ICCV 2007 presentation

Optimizing for weights

empirical loss: $\sum_{i,j,k} [1 - \mathrm{W} \cdot \mathrm{X}_{ijk}]_+$

i,j,k triplets

$\mathrm{W} \cdot \mathrm{X}_{ijk} > 0$

$\mathrm{W} \cdot \mathrm{X}_{ijk} \geq 1$

$$\min_{W,\xi} \frac{1}{2}\|W\|^2 + C \sum_{i,j,k} \xi_{ijk}$$

$$s.t.$$

$$\forall i,j,k : \xi_{ijk} \geq 0$$

$$\forall i,j,k : W \cdot X_{ijk} \geq 1 - \xi_{ijk}$$

$$\forall m : W_m \geq 0$$

A word on duality in optimization

Primal program P:

$$\begin{aligned} &\min\ f(x) \\ &s.t.\quad g(x) \le 0 \\ &\qquad h(x) = 0 \\ &\qquad x \in X. \end{aligned}$$

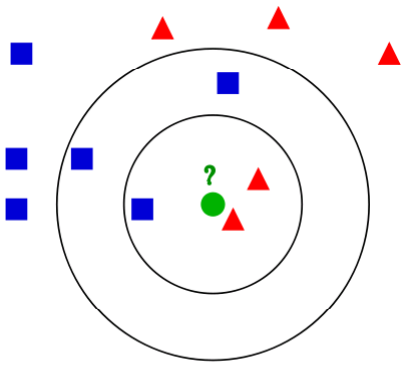
Dual program D:

$$\begin{aligned} &\max\ \Theta(u,v) \\ &s.t.\quad u \ge 0, \\ &where\ \ \Theta(u,v) = \inf\{f(x) + u'g(x) + v'h(x) : x \in X\}. \end{aligned}$$

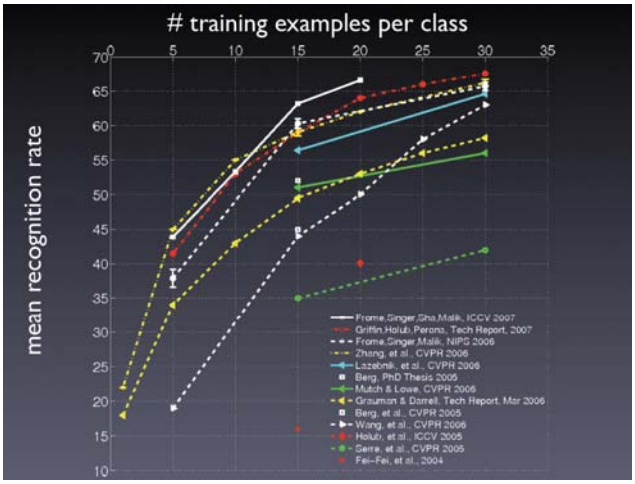
How are the optimal values of the dual and primal programs related?
Weak and strong duality theorem.
Their difference is called the duality gap.

How to categorize with this distance function

- Compute the weights only for a number of training images that represent each category (say 20 images per category)
- When we get a new image we compare it to all the category-representative training images and order the training images based on their distance to the new image.
- Use a 3-NN classifier where if no two images agree on the class within the top 10 matches we take the class of the top-ranked image.

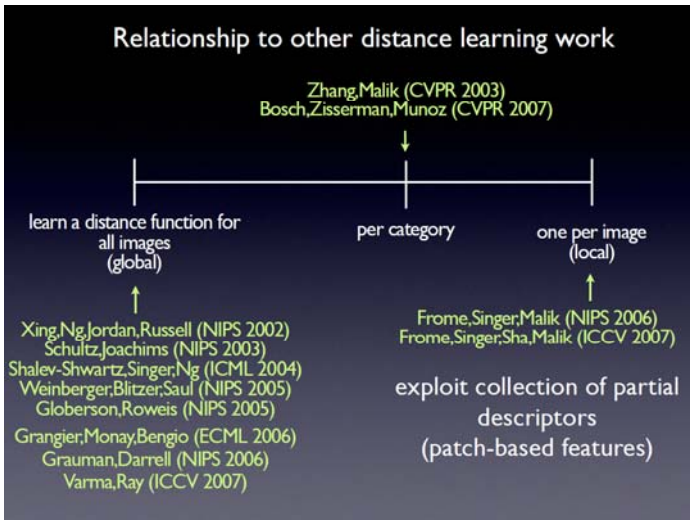


Results



Andrea Frome's ICCV 2007 presentation

Relation to other work

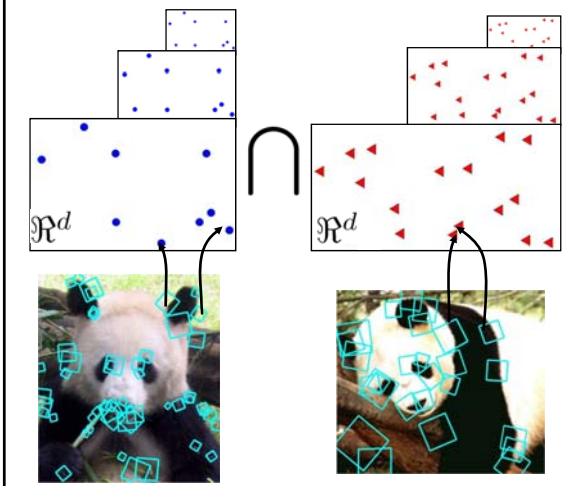


Andrea Frome's ICCV 2007 presentation

Discussion

- Choosing the triplets for training. Too many.
- Choosing the trade-off parameter C.
- Early stopping.
- SVM?
- This method can naturally combine features of very different type e.g. shape features, color features etc.
- The optimization is done on the set of triplets when the actual desired functionality is categorization.
- The duality gap?

The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features, by K. Grauman and T. Darrell. ICCV 2005.



optimal partial matching

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

The challenges

Kernel-based discriminative classification methods can learn complex decision boundaries but there is a problem when:

- Sets of input are unordered
- They vary in cardinality
- And the algorithm needs to be fast

Pyramid match overview

Pyramid match kernel measures similarity of a partial matching between two sets:

- Place multi-dimensional, multi-resolution grid over point sets
- Consider points matched at finest resolution where they fall into same grid cell
- Approximate similarity between matched points with worst case similarity at given level

The following slides are from Kristen Grauman's ICCV 2005 presentation

Pyramid match

Approximate partial match similarity

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

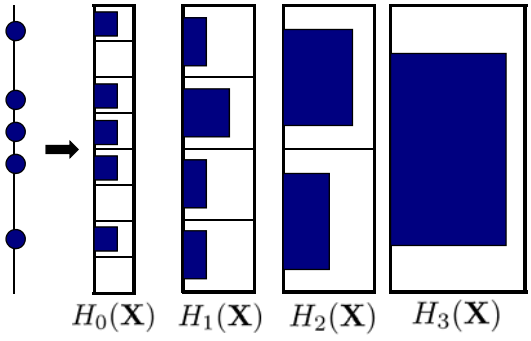
Number of newly matched pairs at level *i*

Measure of difficulty of a match at level *i*

[Grauman and Darrell, ICCV 2005]

Pyramid extraction

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



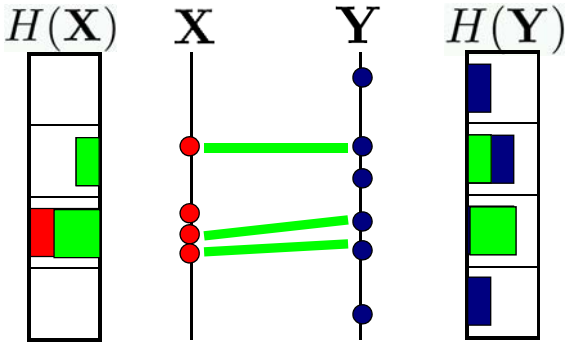
Histogram pyramid: level *i* has i bins of size 2^i

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

Counting matches

Histogram intersection

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



$$\mathcal{I}(H(\mathbf{X}), H(\mathbf{Y})) = 3$$

Counting new matches

Histogram intersection

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

matches at this level — matches at previous level

$$N_i = \mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)$$

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

Pyramid match

histogram pyramids

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

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

measure of difficulty of a match at level *i*

- For similarity, weights inversely proportional to bin size
- Normalize kernel values to avoid favoring large sets

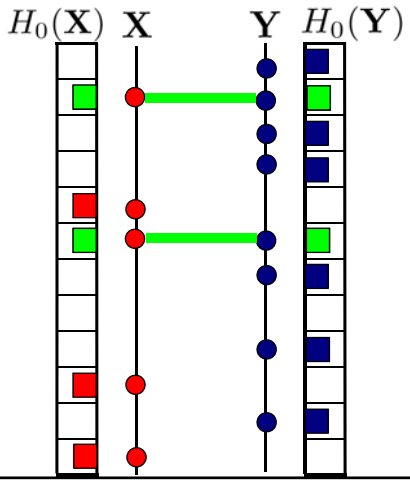
Efficiency

Pyramid match complexity $O(dmL)$

d	feature dimension
m	set size
$L = \log(D)$	number of pyramid levels
D	range of feature values

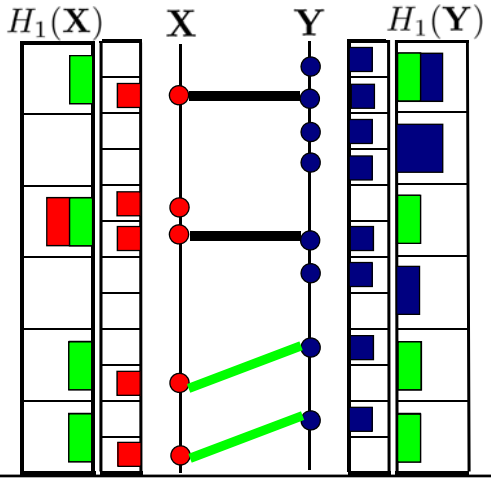
Example pyramid match

$$\mathcal{I}\left(H_0(\mathbf{X}), H_0(\mathbf{Y})\right) = 2 \longrightarrow \begin{matrix} N_0 = 2 \\ w_0 = 1 \end{matrix}$$



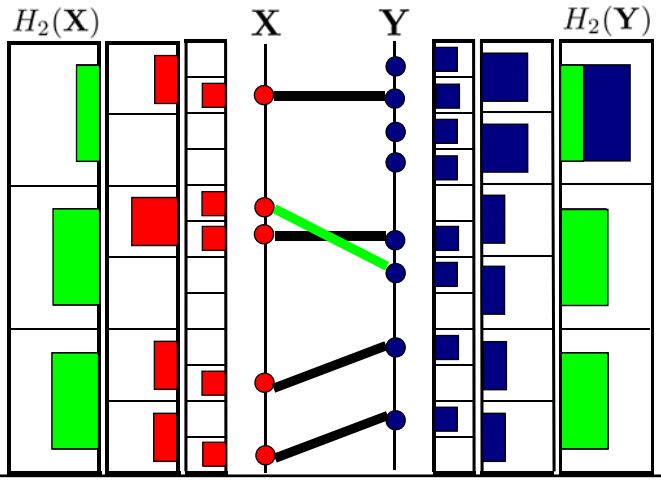
Example pyramid match

$$\mathcal{I}\left(H_1(\mathbf{X}), H_1(\mathbf{Y})\right) = 4 \longrightarrow \begin{matrix} N_1 = 4 - 2 = 2 \\ w_1 = \frac{1}{2} \end{matrix}$$



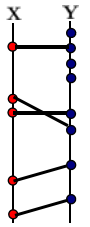
Example pyramid match

$$\mathcal{I}\left(H_2(\mathbf{X}), H_2(\mathbf{Y})\right) = 5 \longrightarrow \begin{array}{l} N_2 = 5 - 4 = 1 \\ w_2 = \frac{1}{4} \end{array}$$



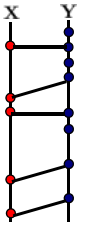
Example pyramid match

pyramid match



$$K_\Delta = \sum_{i=0}^{L} w_i N_i$$
$$= 1(2) + \tfrac{1}{2}(2) + \tfrac{1}{4}(1) = 3.25$$

optimal match



$$K = \max_{\pi:\mathbf{X}\to\mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$
$$= 1(2) + \tfrac{1}{2}(3) = 3.5$$

Mercer's Condition

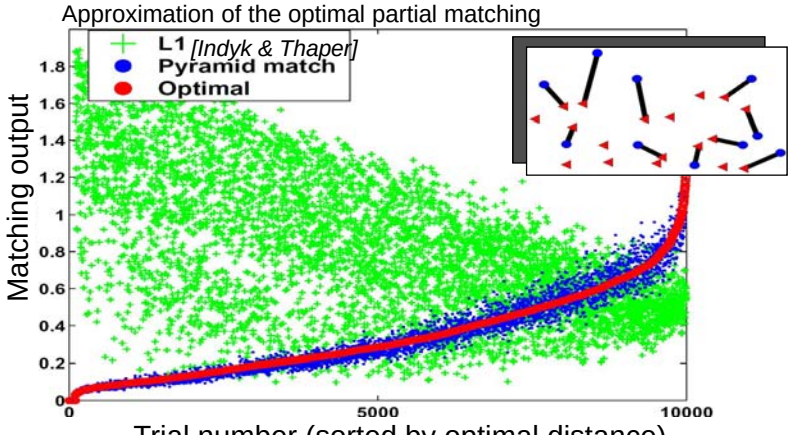
- Such a condition means that there exists a mapping to a reproducing kernel Hilbert space (a Hilbert space is a vector space closed under dot products) such that the dot product there gives the same value as the kernel function.
- The positive definiteness of the kernel would guarantee the convergence of SVM's optimization.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle, \ \forall \mathbf{x}_i, \mathbf{x}_j \in X,$$

$$K_\Delta\left(\Psi(\mathbf{y}), \Psi(\mathbf{z})\right) = \frac{\min(|\mathbf{y}|, |\mathbf{z}|)}{2^L} + \sum_{i=0}^{L-1} \frac{1}{2^{i+1}} \mathcal{I}\left(H_i(\mathbf{y}), H_i(\mathbf{z})\right),$$

Optimal partial matching

Approximation of the optimal partial matching



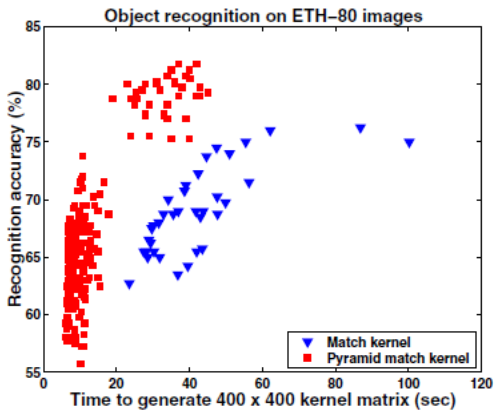
100 sets with 2D points, cardinalities vary between 5 and 100

Grauman and Darrel ICCV 2005

How to build a classifier with this kernel

- Train an SVM by computing kernel values between all labeled training examples
- Classify novel examples by computing kernel values against support vectors
- One-versus-all for multi-class classification
- Since the Kernel is positive definite, convergence is guaranteed.

Recognition results

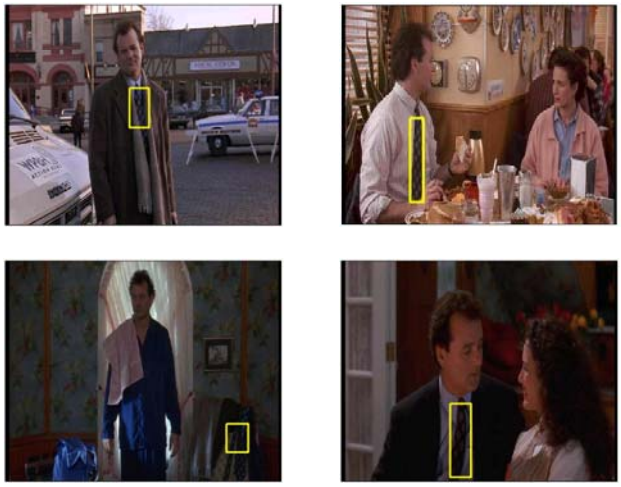


Grauman and Darrel ICCV 2005

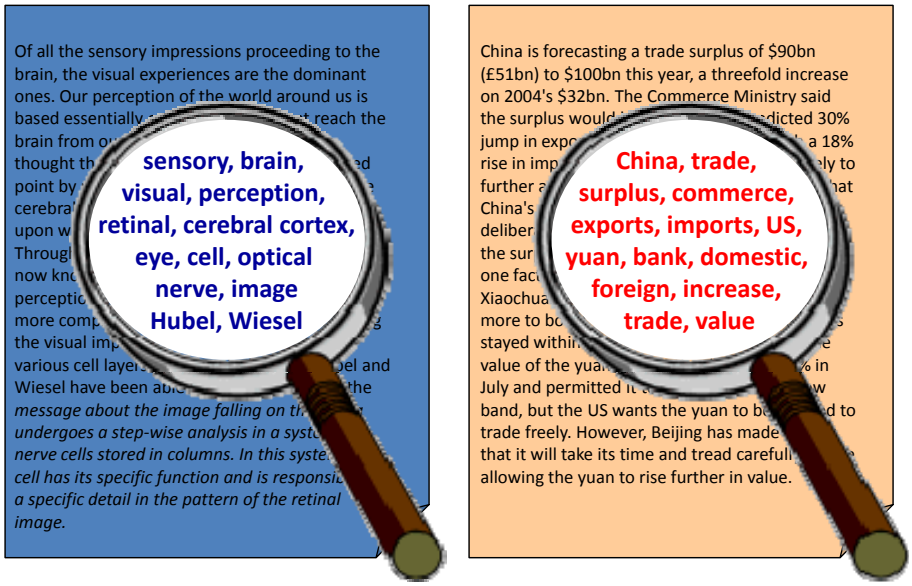
Features of pyramid kernel method

- linear time complexity
- no independence assumption
- model-free
- insensitive to clutter
- positive-definite function
- fast, effective object recognition $O(dmL)$

Video Google: A Text Retrieval Approach to Object Matching in Videos, J. Sivic and A. Zisserman, 2003.



Analogy to documents

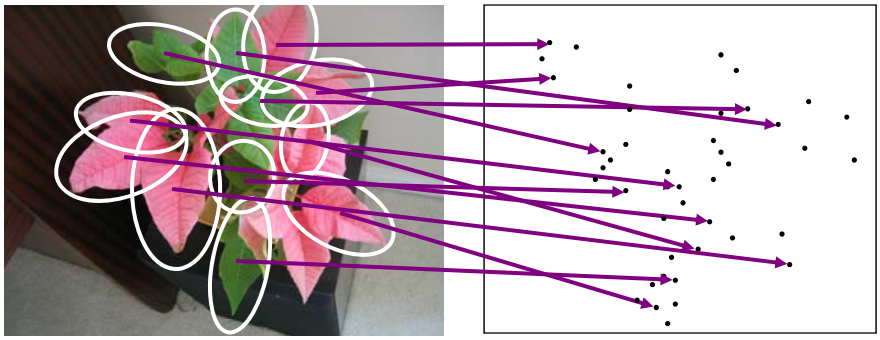


ICCV 2005 short course, L. Fei-Fei

Matching features in different views

Sivic and Zisserman 2003

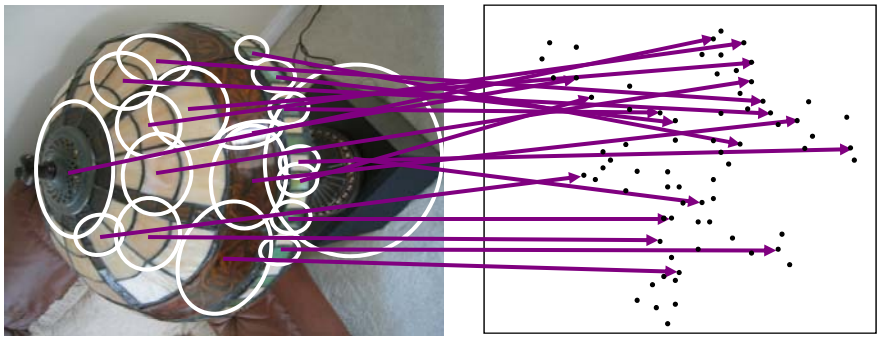
Visual words: main idea



Slide credit: D. Nister

K. Grauman, B. Leibe

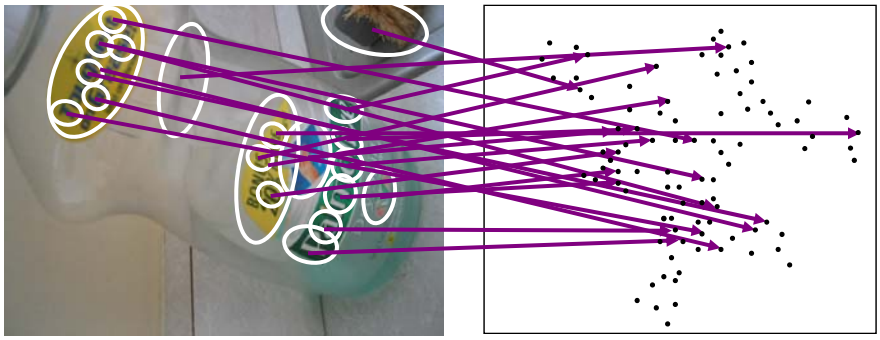
Visual words: main idea



Slide credit: D. Nister

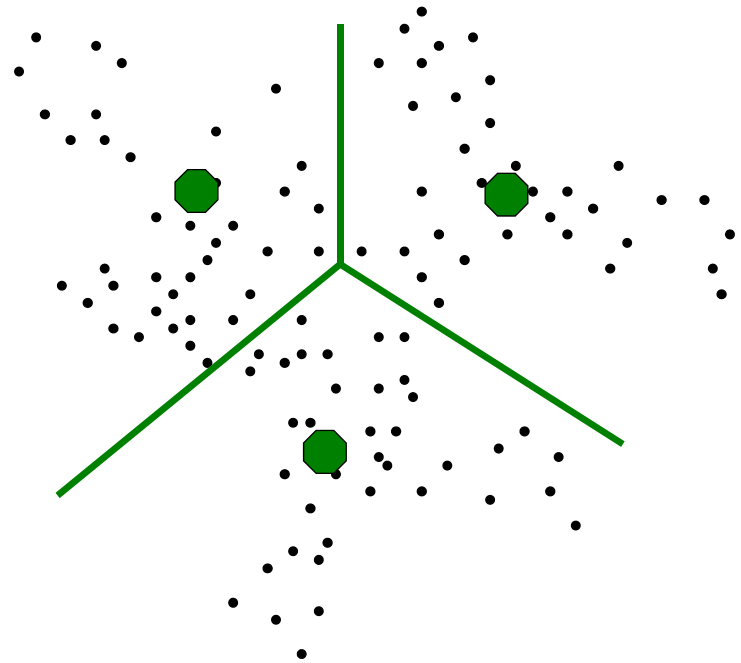
K. Grauman, B. Leibe

Visual words: main idea



Slide credit: D. Nister

K. Grauman, B. Leibe



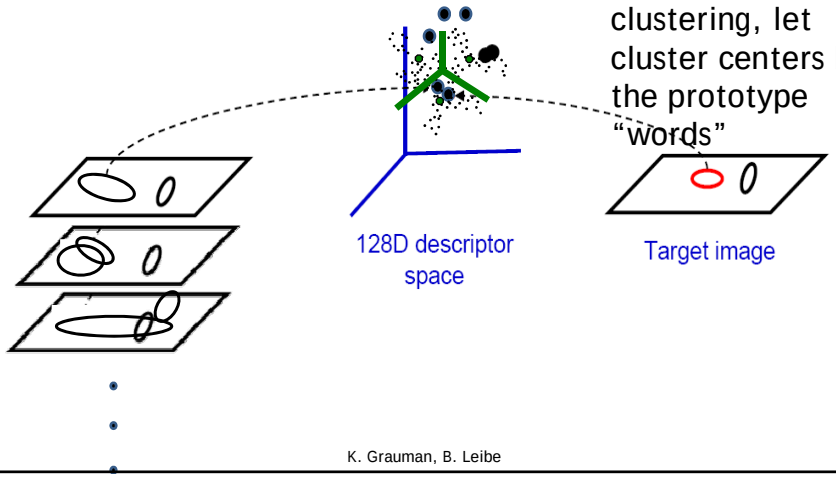
Slide credit: D. Nister

K. Grauman, B. Leibe

Visual words: main idea

Map high-dimensional descriptors to

- Quantize via clustering, let cluster centers be the prototype "words"



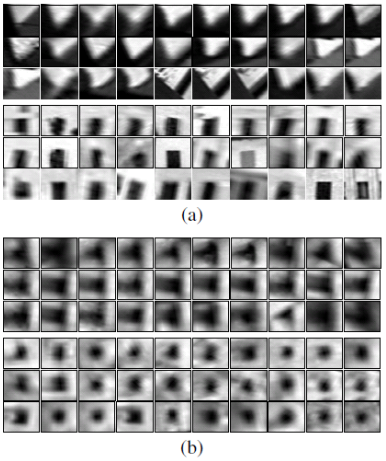
K. Grauman, B. Leibe

Clusters of visual words

The descriptors are vector quantized into clusters using K-means clustering.

K-means is run several times with random initial conditions and the best one is chosen.

SA and MS are clustered independently since they cover different and independent regions of the scene.



Indexing local features: inverted file index

- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index…
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we'll need to map our features to "visual words".

K. Grauman, B. Leibe

Inverted file index for images comprised of visual words

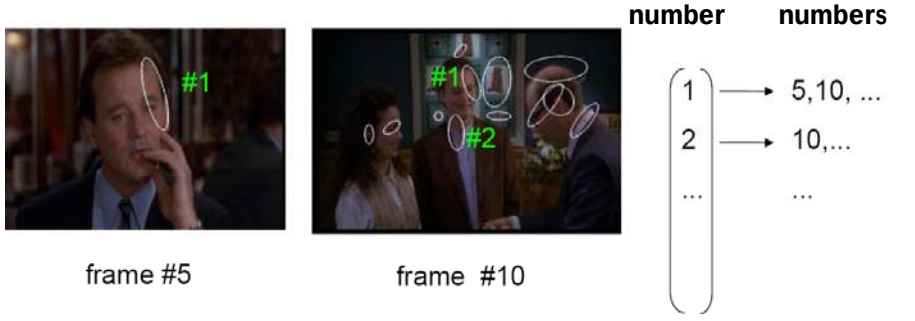
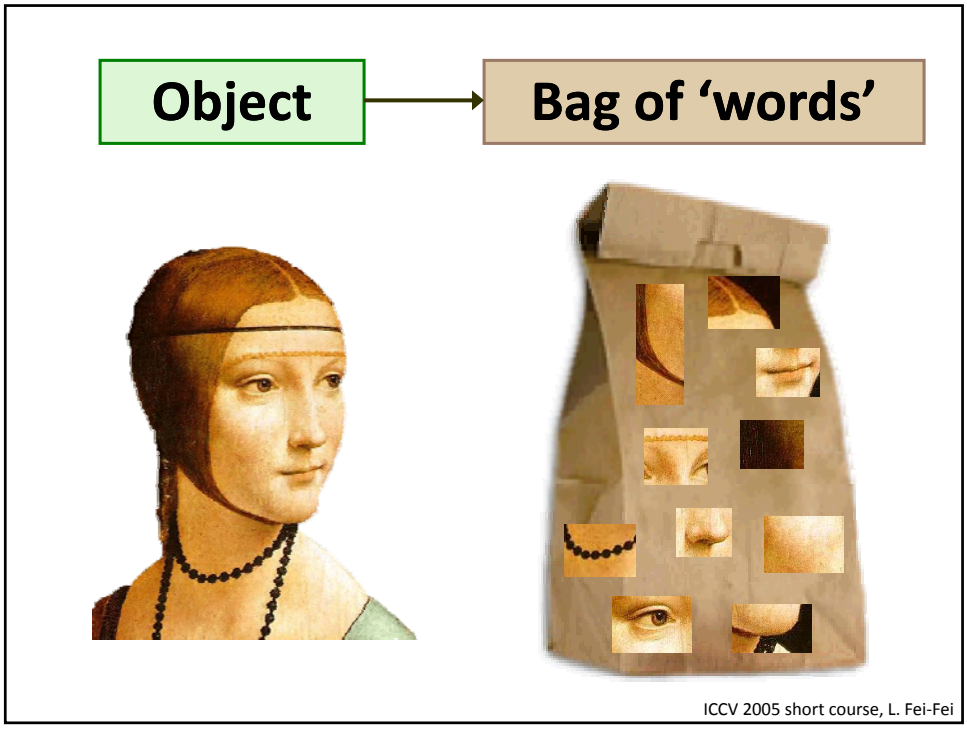
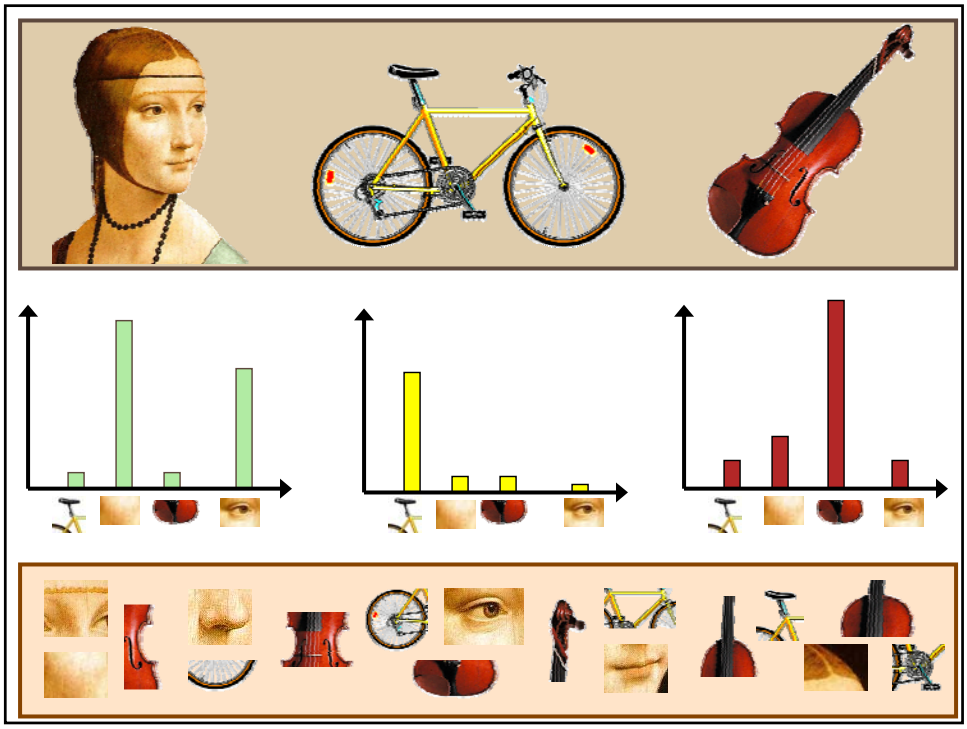


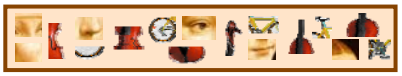
Image credit: A. Zisserman

K. Grauman, B. Leibe


Object
Bag of 'words'
ICCV 2005 short course, L. Fei-Fei



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.

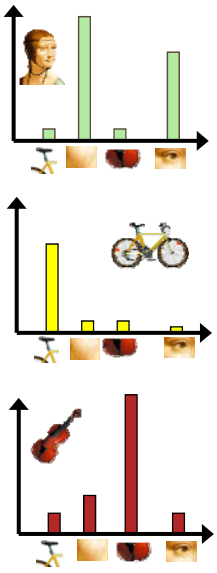


Image credit: Fei-Fei Li

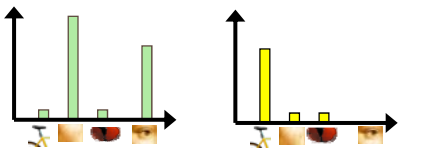
K. Grauman, B. Leibe

51

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]



$\vec{d}_j$ $\vec{q}$

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

tf-idf weighting

- **T**erm **f**requency – **i**nverse **d**ocument **f**requency
- Describe frame by frequency of each word within it, downweight words that appear often in the database
- (Standard weighting for text retrieval)

Number of occurrences of word i in document d

Number of words in document d

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

Total number of documents in database

Number of occurrences of word i in whole database

Bags of words for content-based image retrieval

What if query of interest is a portion of a frame?

Visually defined query

"Groundhog Day" [Rammis, 1993]

"Find this clock"

"Find this place"

Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

Example

Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

Discussion

- The use of video information
- Stop list
- Spatial consistency
- Categorization / recognition?
- Where is the distance function?
- Alternative to sliding window!

The big picture

In what ways are these methods:

- Similar?
- Different?
- Related?

Similar issues

- The three papers discussed here deal with the same kind of problem which is finding a measure for the visual similarity of images.
- They all base their methods on the previously extracted feature descriptors in an image.
- Some of the benefits of working with features is that it makes the algorithm robust to clutter, noise, background (irrelevant information in general) as well as making partial search possible.
- The spatial information is generally ignored (save for a brief mention in the video google paper) so if you shuffle the features in an image you will get a very similar image by these measures.

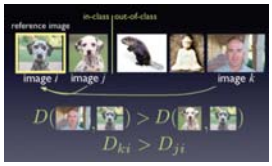
Differences

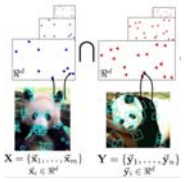
Each method is tuned for a slightly different application.

- The Frome method is designed mostly for categorization. The big advantage is that the weights can emphasize important features.
- The pyramid kernel is defines a kernel that can be used as the core in different methodologies. Probably the most compatible of all with different algorithm.
- The video google generalized a text retrieval system for fast image search.

Globally-consistent vs pyramid kernel

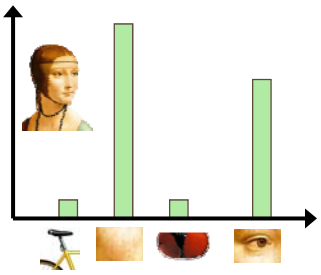
- Frome claims better performance however her method is tuned for that special task.
- The Frome distance can easily incorporate very different types of features as the distances are computed independently (Compute multiple pmk matrices, add them, or add weighted matrices).
- The weights make the Frome distance more robust to irrelevant information (in general).
- The distance defined in Frome is not a real mathematical distance so it has limited use elsewhere. Pyramid kernel is positive definite measure of distance thus compatible with SVM.
- The pyramid kernel is much faster.
- The pyramid kernel has less parameters to tune (could be good or bad).





Video google vs the other two

- The video google is extremely fast for image retrieval but requires a long preprocessing (building the indexing file).
- It is however less accurate for categorization and object recognition.
- can we have "universal" vocabulary independent of dataset?
- single level vocabulary of Video Google vs multi-resolution vocabulary implied by the pmk.



Thank you!