Efficient Matching for Retrieval and Recognition

Kristen Grauman
Trevor Darrell
MIT

DOE CSGF
Annual Fellows' Conference
June 22, 2006

Recognition

Wild card

Tower Bridge

Steve Carell

Specific

butterfly

butterfly

building

building

Categories

Key challenges: robustness

Illumination

Object pose

Clutter

Occlusions

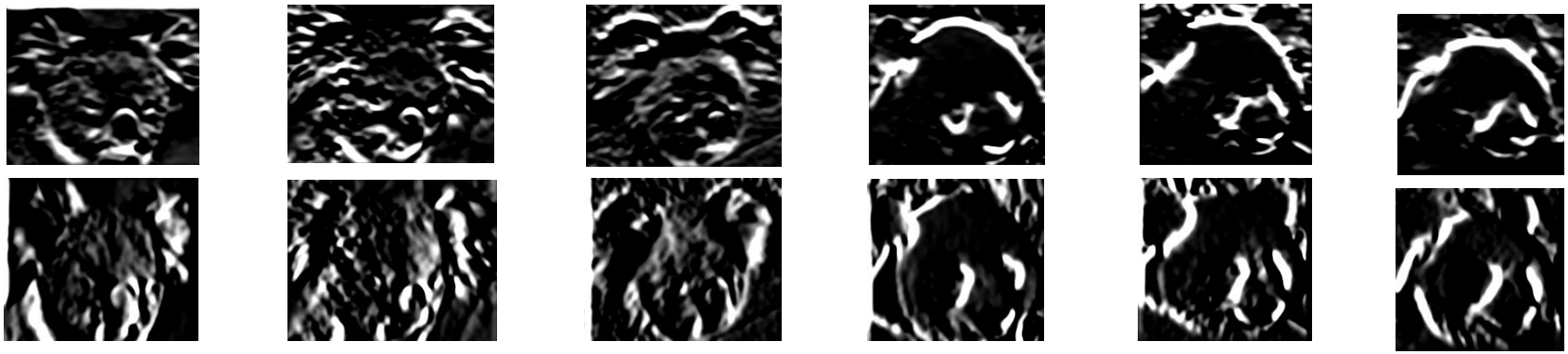
Intra-class appearance

Viewpoint

Key challenges: efficiency

- Thousands to millions of pixels in an image
- 3,000-30,000 human recognizable object categories
- 30+ degrees of freedom in the pose of articulated objects (humans)
- Billions of images indexed by Google Image Search
- 18 billion+ prints produced from digital camera images in 2004
- 295.5 million camera phones sold in 2005

Global image representations



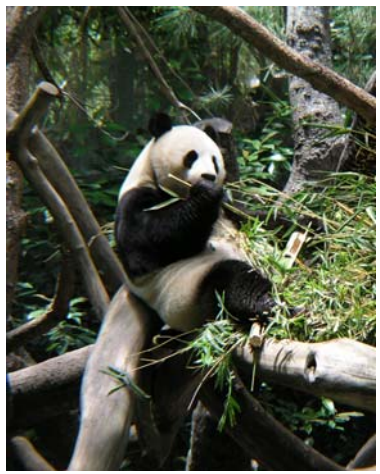
Map image to a single vector based on overall characteristics

- vector of pixel intensities
- grayscale / color histogram
- bank of filter responses …

Limitations of global representations

- Success may rely on alignment
- All parts of image impact description

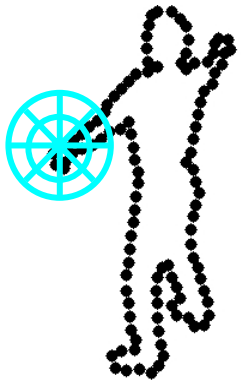




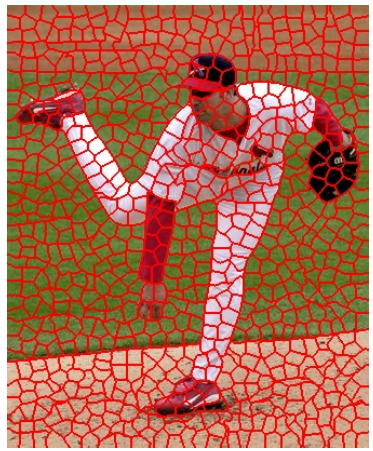

Local image representations

Describe component regions or patches separately

SIFT [Lowe]

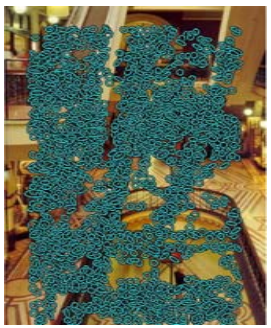


Shape context
[Belongie et al.]

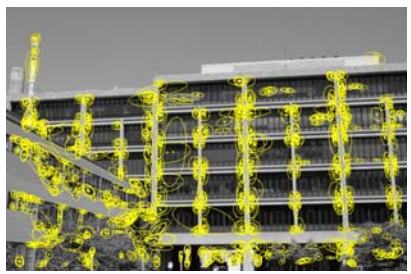


Superpixels
[Ren et al.]

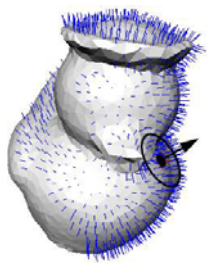
Maximally Stable Extremal Regions [Matas et al.]



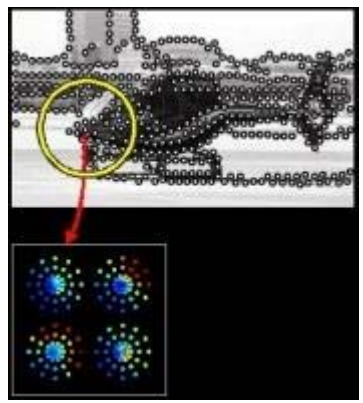
Salient regions
[Kadir et al.]



Harris-Affine
[Schmid et al.]



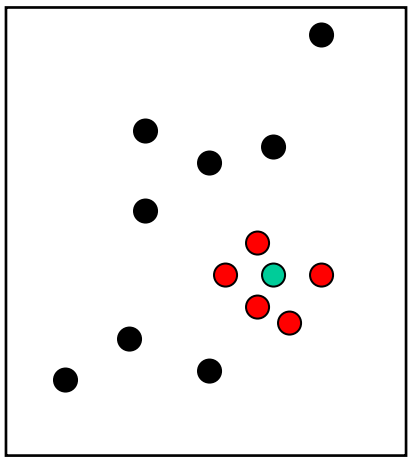
Spin images
[Johnson and Hebert]



Geometric Blur
[Berg et al.]

Comparing images

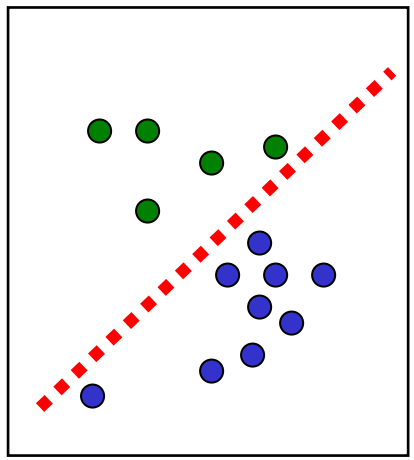
A meaningful way of comparing two image descriptions is key for various tasks:



Nearest neighbor

↓

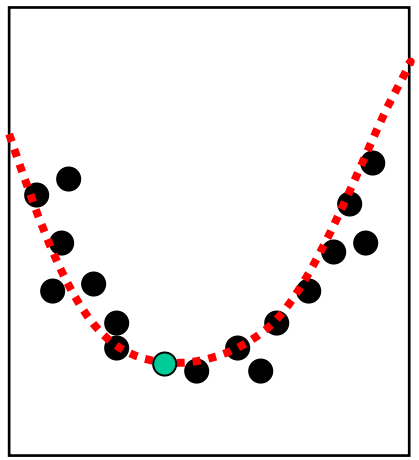
Image retrieval



Classifying

↓

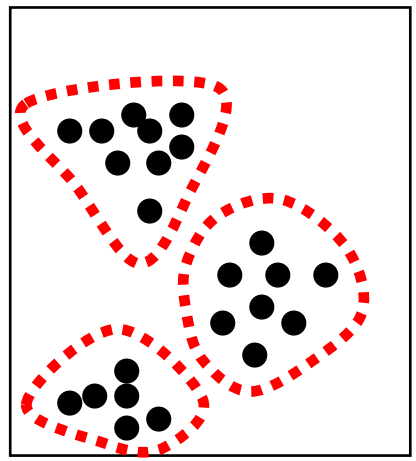
Object recognition



Regression

↓

Pose inference



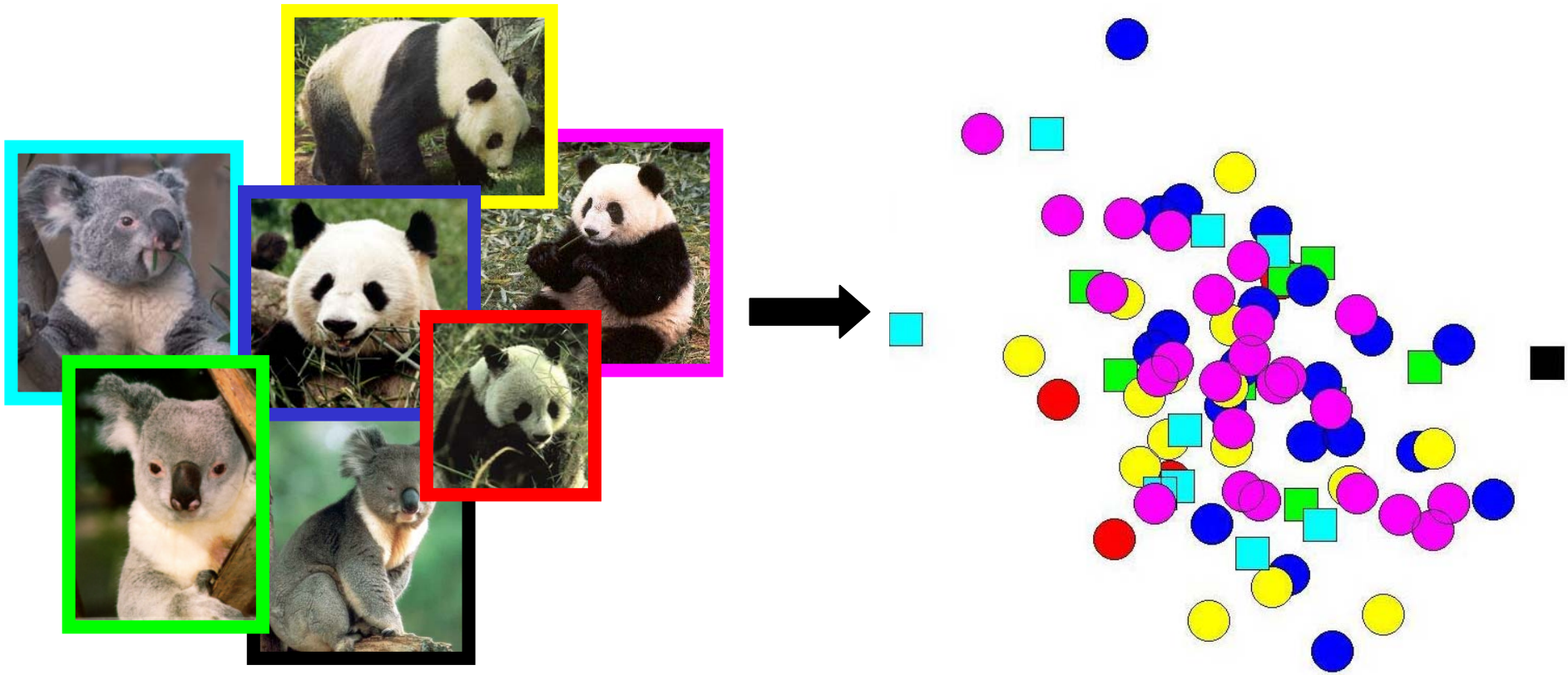
Clustering

↓

Category discovery

How to handle sets of features?

- Each instance is unordered set of vectors
- Varying number of vectors per instance



Previous approaches

- Voting
  [Schmid, Lowe, Tuytelaars et al.,…]
- Fit (parametric) distributions to sets
  [Kondor & Jebara, Moreno et al., Cuturi & Vert, Wolf & Shashua]
- Compare histograms over prototype features
  [Csurka et al., Sivic & Zisserman, Lazebnik & Ponce, Agarwal & Triggs]
- Explicit search for correspondences
  [Wallraven et al., Lyu, Boughhorbel et al., Belongie et al., Rubner et al., Berg et al., Gold & Rangarajan, Shashua & Hazan,…]
- Multi-resolution matching approximations
  [Indyk & Thaper, Charikar, Agarwal & Varadarajan…]

Limitations

Costly comparisons

Costly training

Sensitivity to clutter

Restrictive assumptions

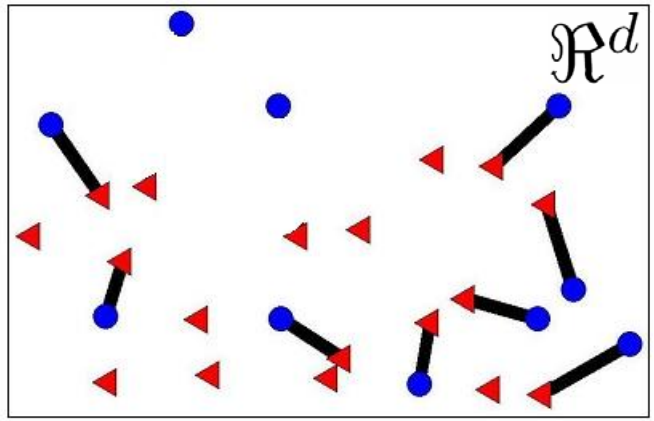
Assumes feature independence

Restricted to sets of equal sizes

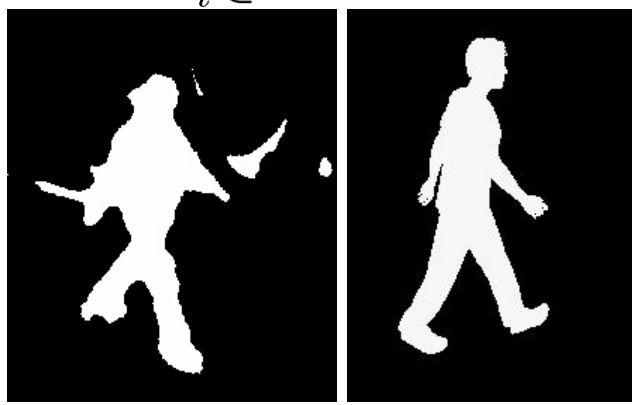
Inapplicable for discriminative learning

Partial matching

Compare sets by computing a *partial matching* between their features.



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



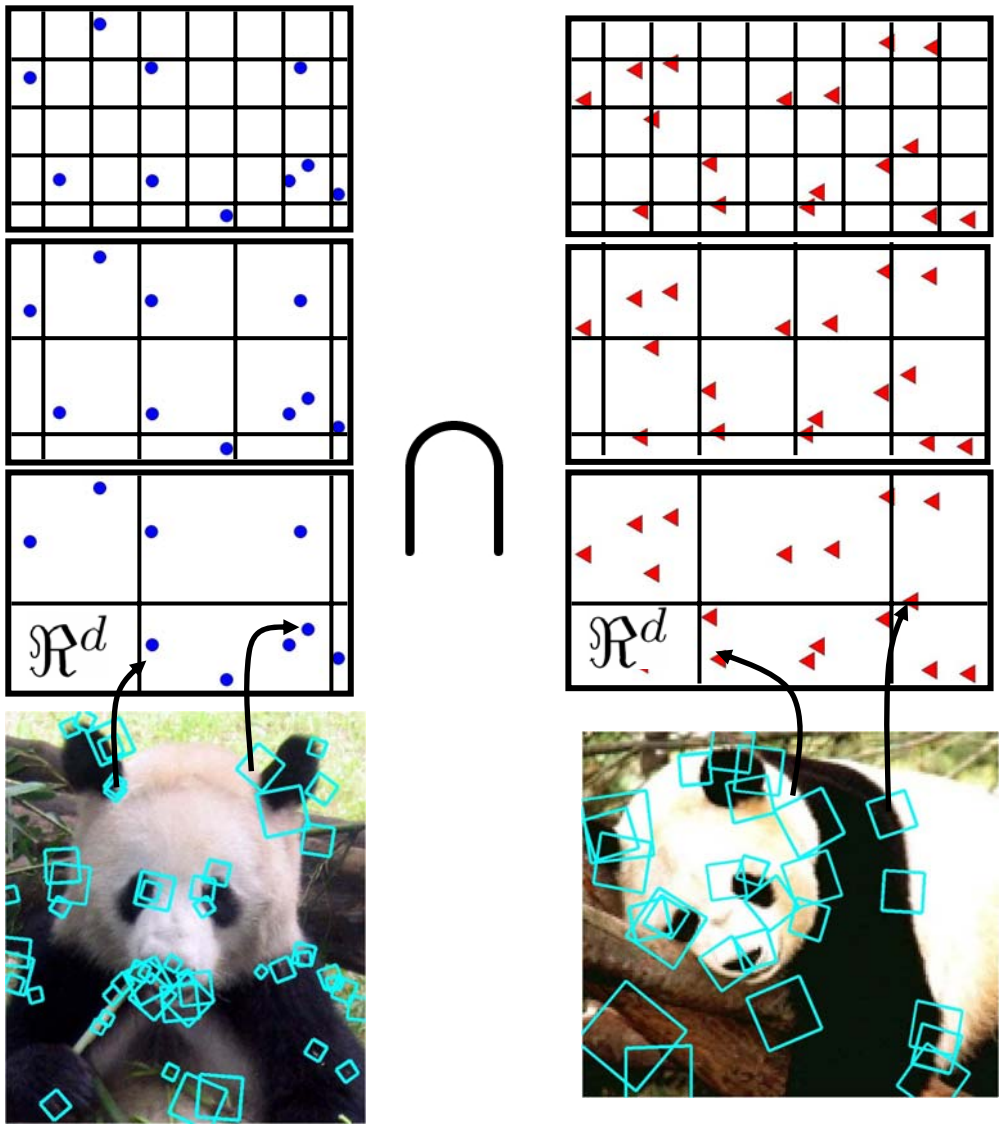
Computing the partial matching

- Optimal matching $O(m^3)$
- Greedy matching $O(m^2 \log m)$
- Pyramid match $O(mL)$

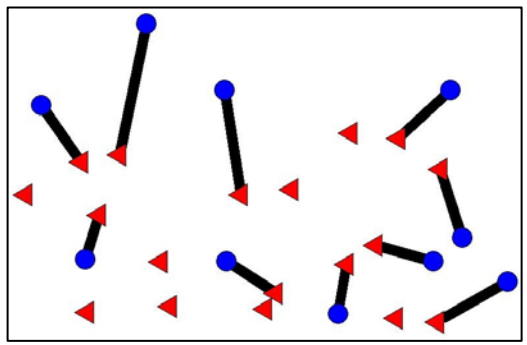
  [Grauman and Darrell, ICCV 2005]

for sets with $O(m)$ features

Pyramid match overview



$\approx$



optimal partial matching

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

$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Pyramid match overview

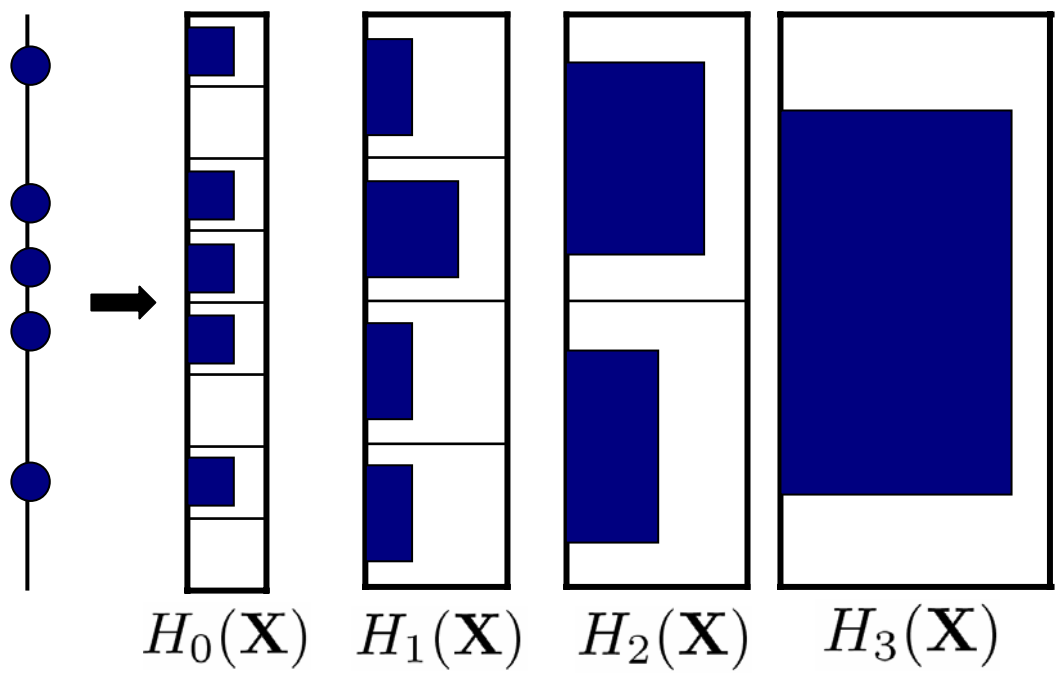
Pyramid match measures similarity of a partial matching between two sets:

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

No explicit search for matches!

Pyramid extraction

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



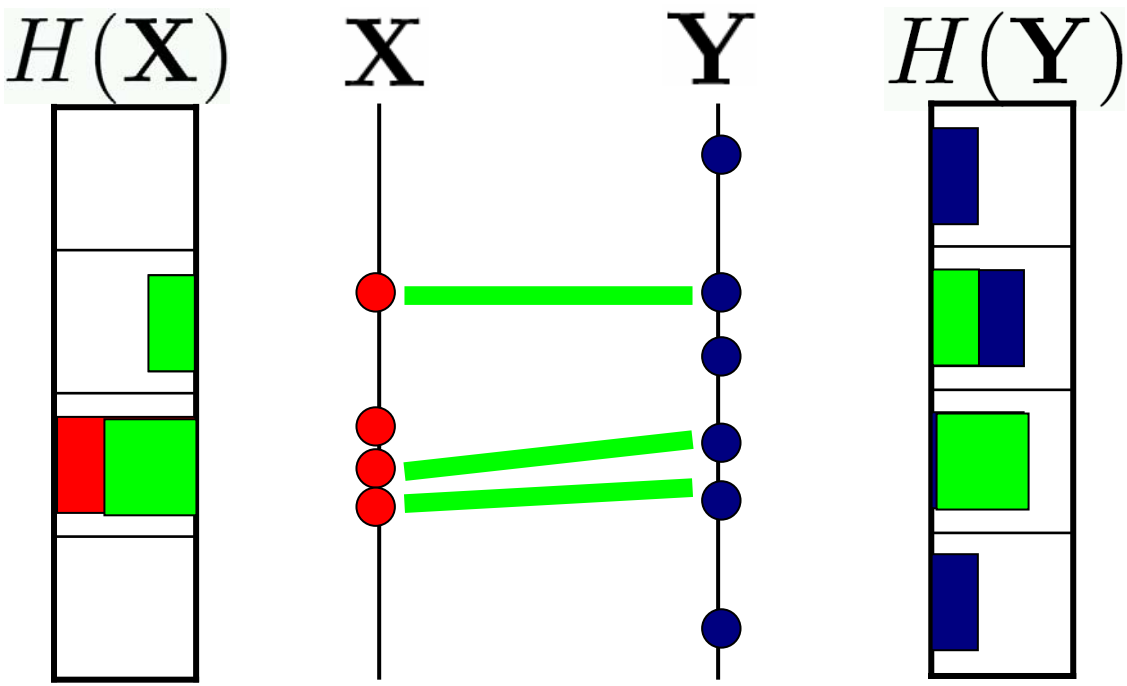
Histogram pyramid: level i has bins of size 2^i

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

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

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

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

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

Pyramid match

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

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

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

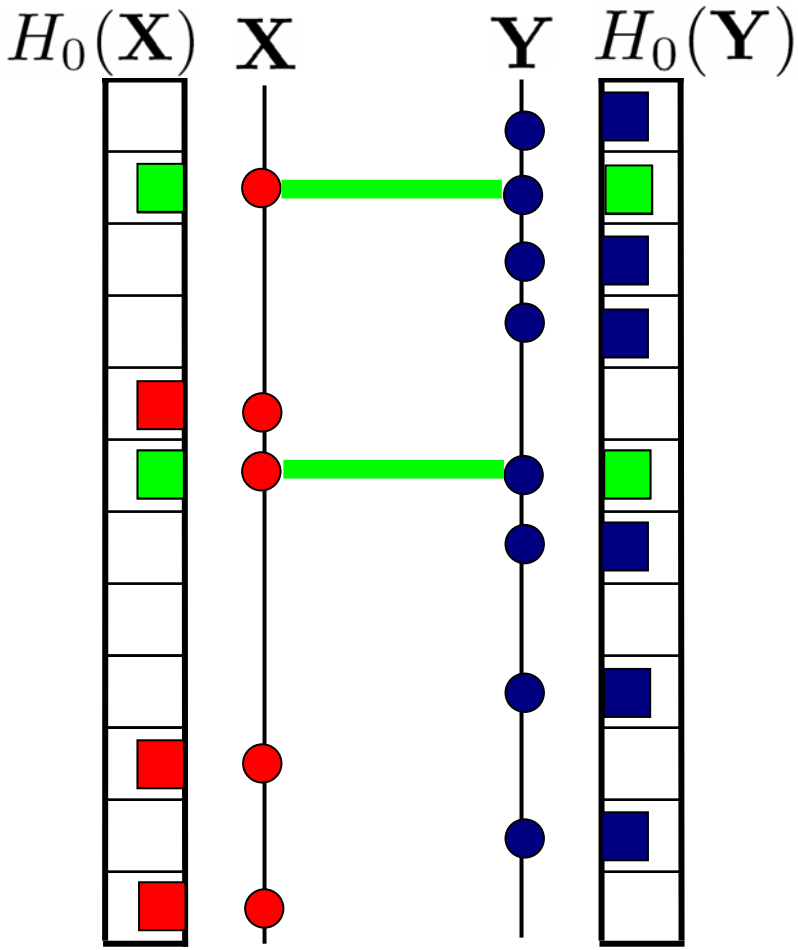
Efficiency

Pyramid match complexity $O(dmL)$

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

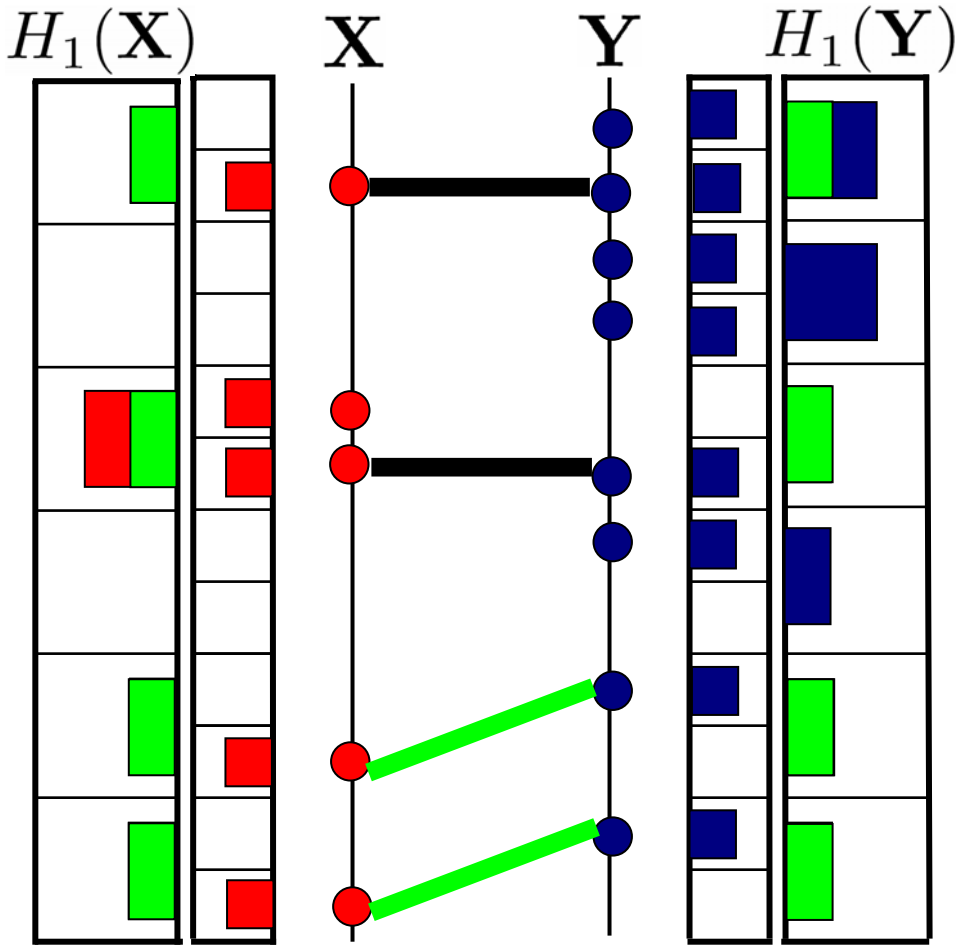
Example pyramid match

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



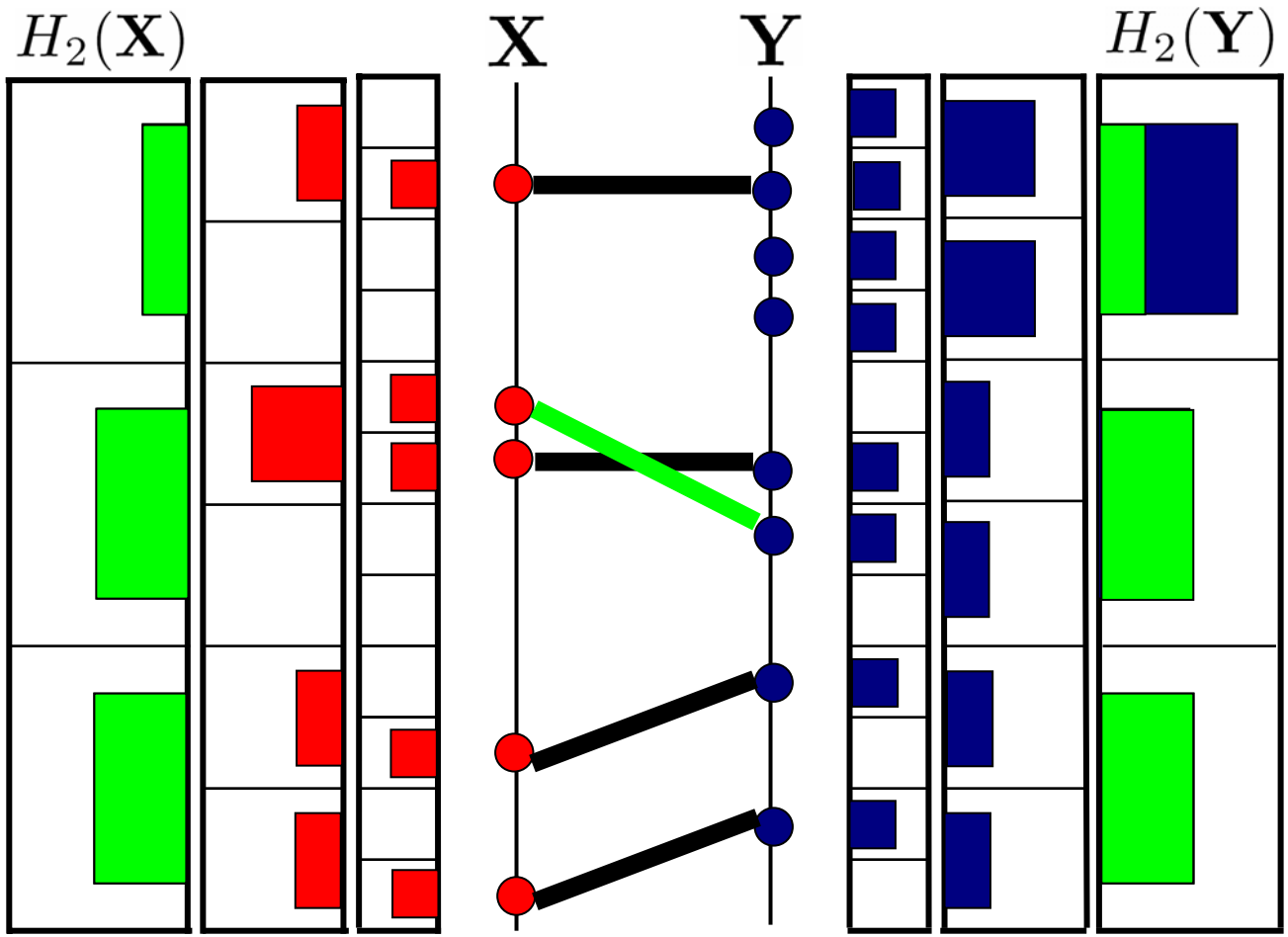
Example pyramid match

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



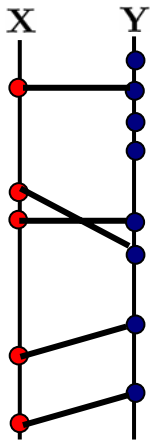
Example pyramid match

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



Example pyramid match

pyramid match

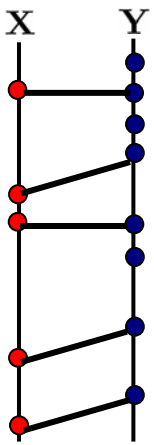


$$K_\Delta = \sum_{i=0}^{L} w_i N_i$$

$$= 1(2) + \tfrac{1}{2}(2) + \tfrac{1}{4}(1) = 3.25$$

---

optimal match

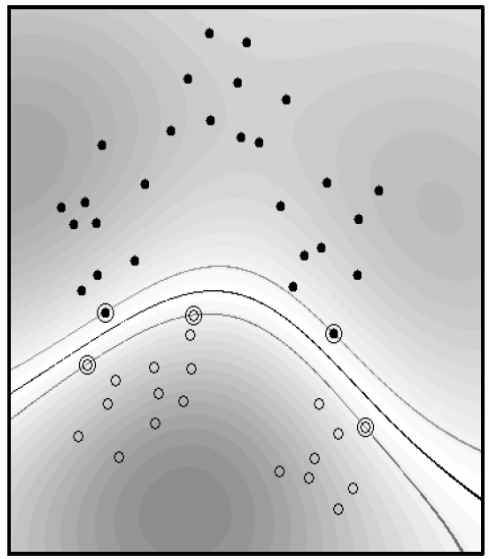


$$K = \max_{\pi:\mathbf{X}\rightarrow\mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

$$= 1(2) + \tfrac{1}{2}(3) = 3.5$$

Learning with the pyramid match

- Kernel-based methods
  - Embed data into a Euclidean space via a similarity function (kernel), then seek linear relationships among embedded data
  - Include classification, regression, clustering, dimensionality reduction,…
  - Efficient and good generalization

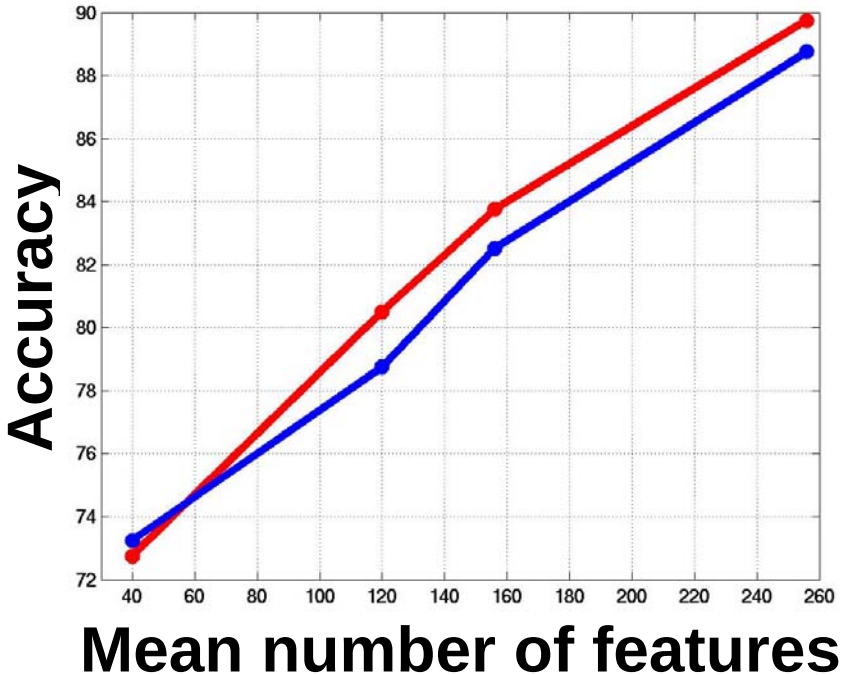


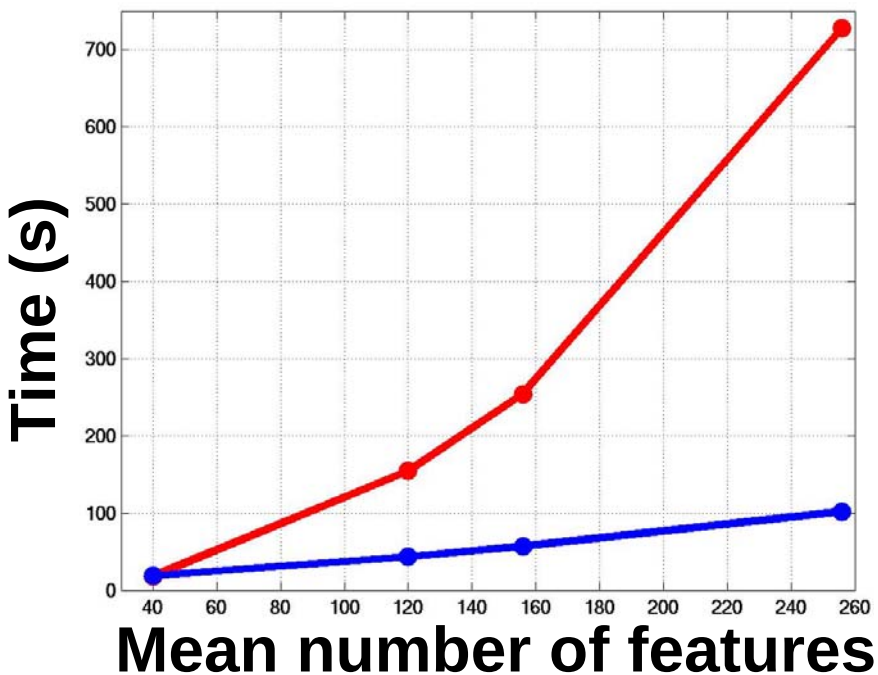
- Pyramid match forms a Mercer kernel

Category recognition results

Kernel	Complexity
Match *[Wallraven et al.]*	$O(dm^2)$
Pyramid match	$O(dmL)$

ETH-80 data set

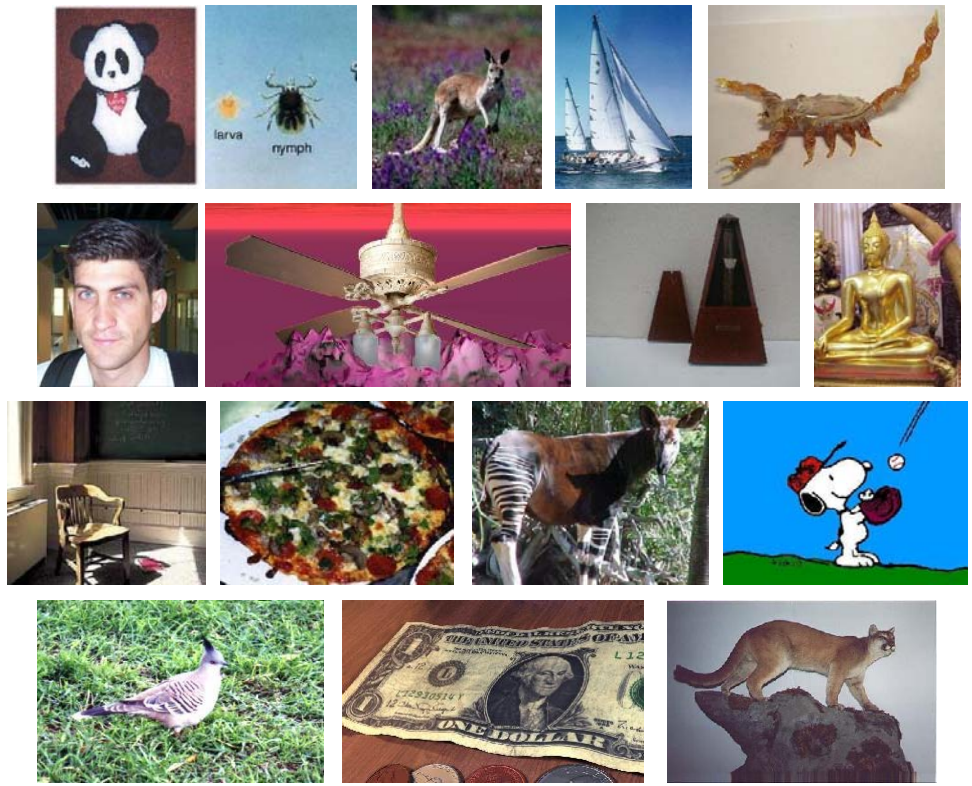




Category recognition results

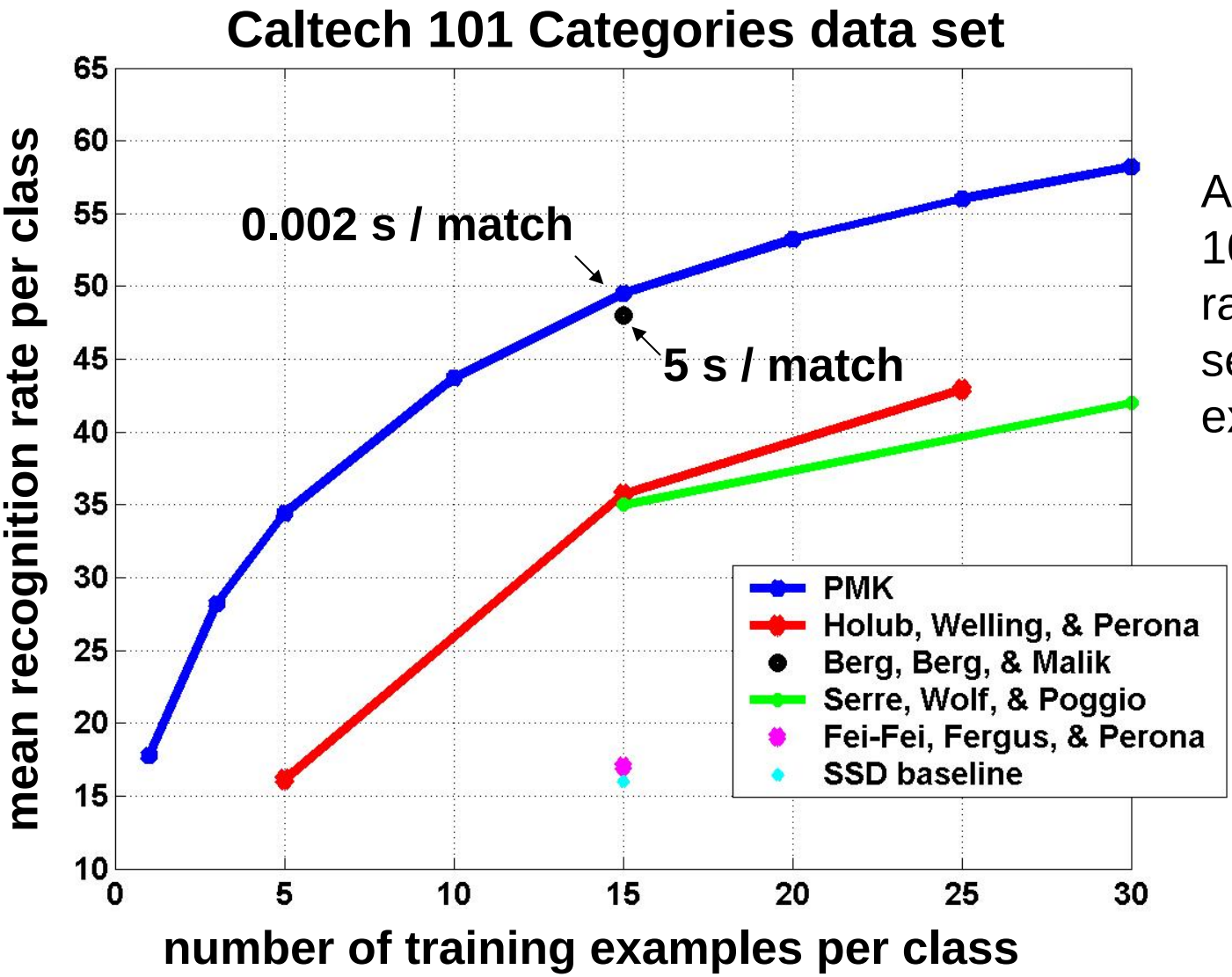
- 101 categories
  40-800 images per class
  8677 total images
- One-vs-all SVM with PMK
- Features:
  - Densely sampled
  - SIFT descriptor + spatial
  - Average *m*=1140 per set

Caltech 101 data set



Data provided by Fei-Fei Li, Rob Fergus and Pietro Perona

Category recognition results



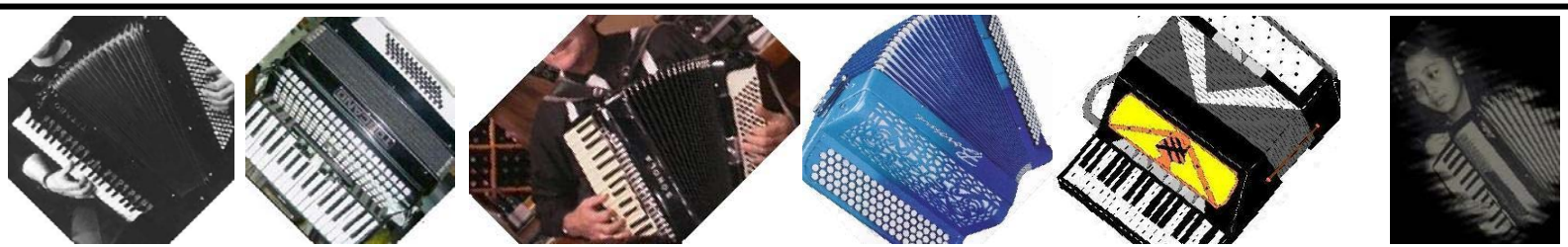
Averaged over 10 runs with randomly selected training examples

Our 5 best categories

100%

100%

99.7%



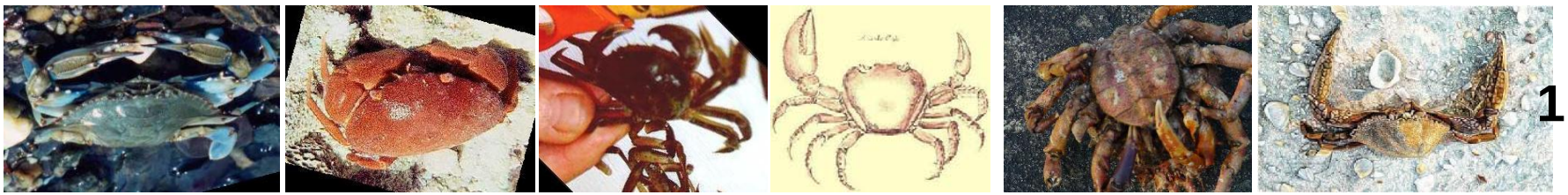
99.1%

98.2%

Our 5 worst categories

7.7%

11.2%

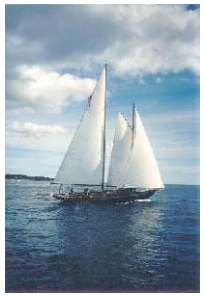


11.5%

11.8%

12.3%

Most confused category pairs




schoo·ner *n.* **schooner**

A fore-and-aft rigged sailing vessel having at least two masts, with a foremast that is usually smaller than the other masts.





ketch *n.* *Nautical* **ketch**

A two-masted fore-and-aft-rigged sailing vessel with a mizzenmast stepped aft of a taller mainmast but forward of the rudder.

Most confused category pairs

lotus

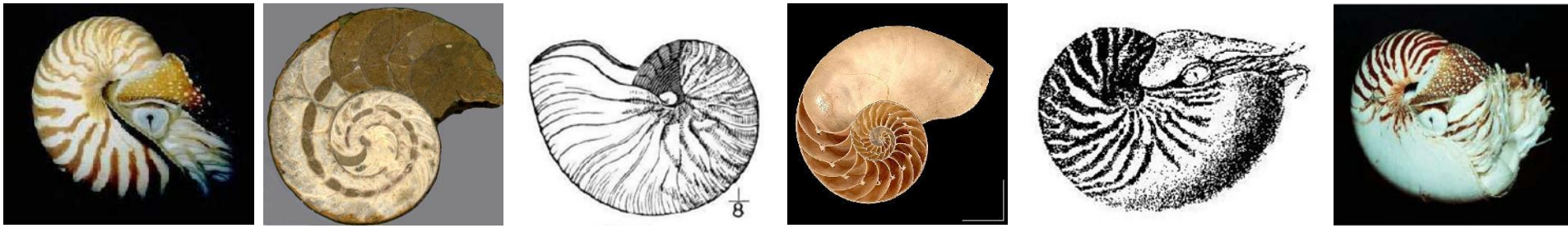
water lily

Most confused category pairs

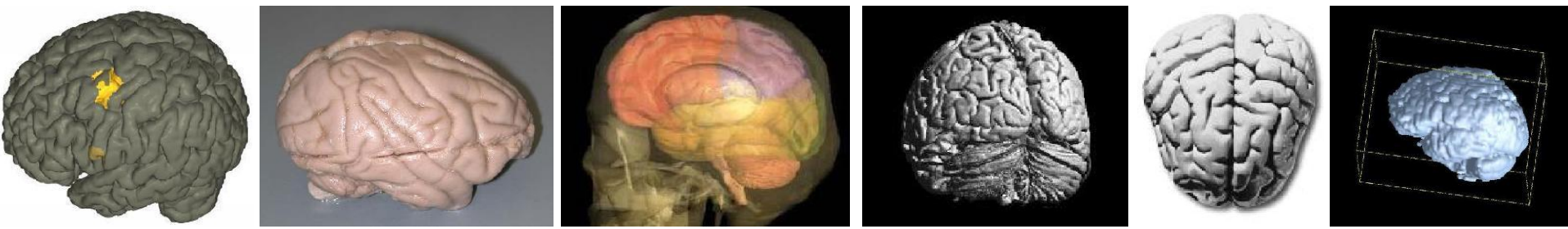
gerenuk

kangaroo

Most confused category pairs



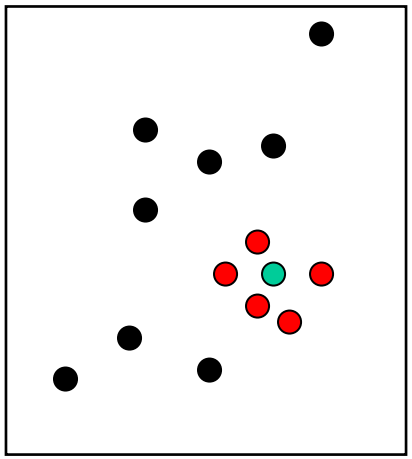
nautilus



brain

Pyramid match applications

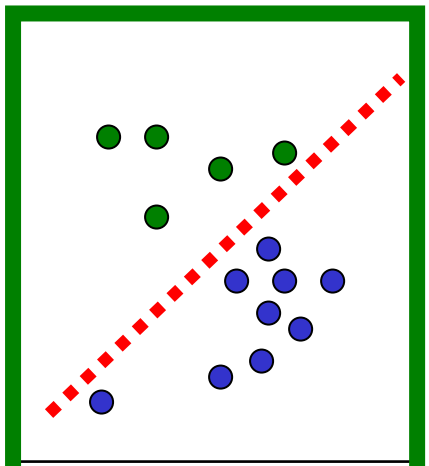
A meaningful way of comparing two image descriptions is key for various tasks:



Nearest neighbor

↓

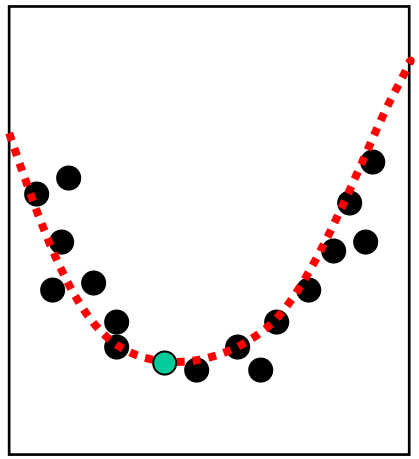
Image retrieval



Classifying

↓

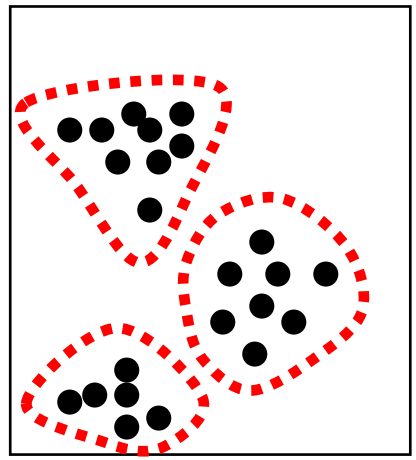
Object recognition



Regression

↓

Pose inference



Clustering

↓

Category discovery

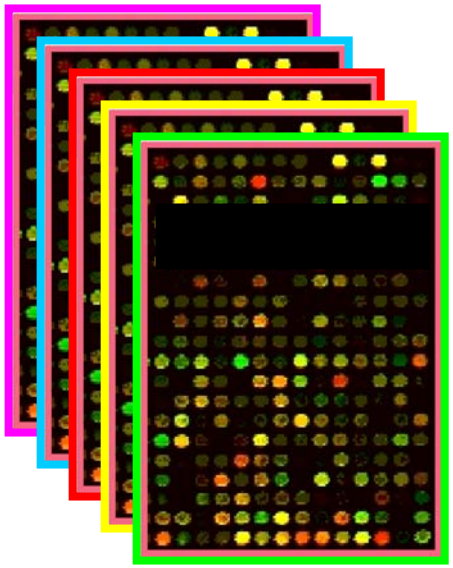
Contributions

Pyramid match: a new similarity measure over sets of vectors that efficiently forms an implicit partial matching

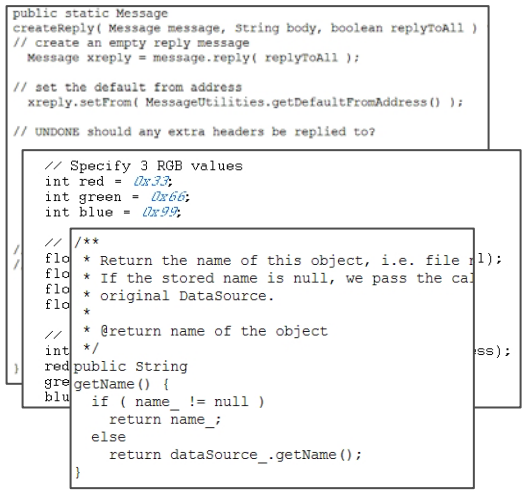
- linear time complexity
- no independence assumption
- model-free
- insensitive to clutter
- positive-definite function (a **kernel**)

Demonstrated effectiveness for retrieval, recognition, and regression tasks with local image features

Sets of features elsewhere



diseases as sets of gene expressions



methods as sets of instructions

documents as bags of words

Acknowledgements

Trevor Darrell

MIT Vision Interface Group

DOE Computational Science
Graduate Fellowship