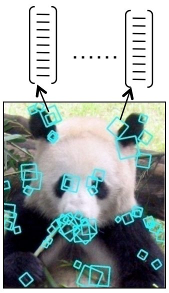
# Pyramid Matching Kernel

## Flow of matching and options so far

Feature detection/sampling

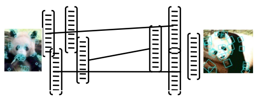
List of positions, scales, orientations



Feature descriptors extraction

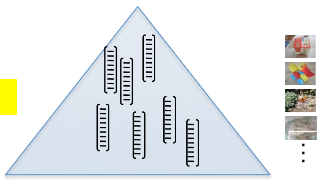
Associated list of d-dimensional descriptors

Opt 1



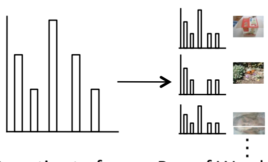
Nearest Neighbor match image to image

Opt 2



Match to an Index of a pool of descriptors from other images

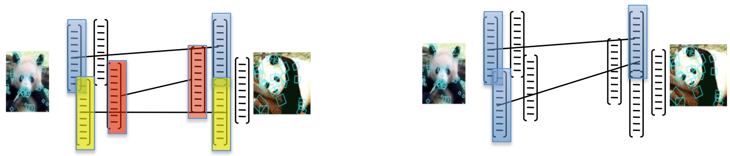
Opt 3



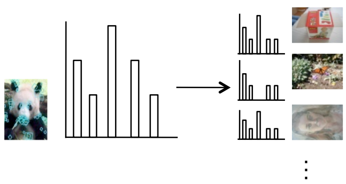
Quantize to form a Bag of Words vector and match to BoWs of other images

Pictures from K. Grauman, B. Leibe

- Nearest neighbour matching :
  - uses each feature in a set to independently index into the second set; this ignores possibly useful information of co-occurrence.
  - fails to distinguish between instances where an object has varying numbers of similar features since multiple features may be matched to a single feature in the other set.



- Bag-of-Words matching:
  - can only compare entire images to one another and does not allow partial matchings.

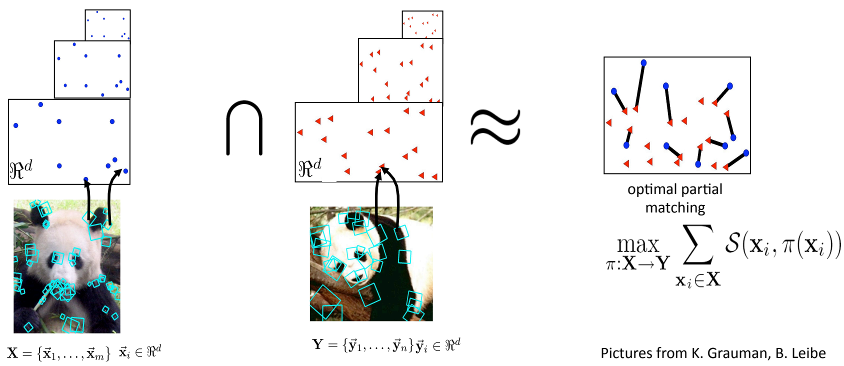


## Pyramid matching

- Pyramid matching is an efficient method by Grauman and Darrell that employs a *multi-resolution histogram pyramid* based on data-dependent partitions of the feature space and *histogram intersection*.

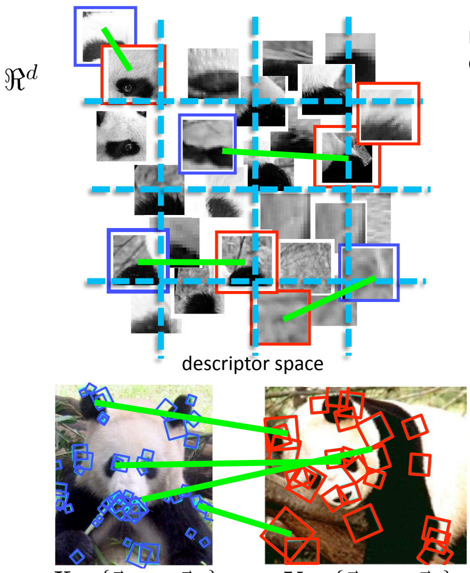
- Pyramid match allows input sets to have unequal cardinalities: enables partial matchings, where the points of the smaller set are mapped to some subset of the points in the larger set.

- With respect to optimal partial matching (minimum of sums of distances between matched points in the lower cardinality set to some subset of the points in the larger set) pyramid matching provides an approximate measure of similarity measure




Pictures from K. Grauman, B. Leibe

## Feature space pyramid partitions



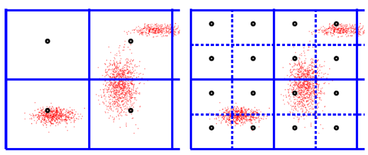

$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Feature space partitions serve to match the local descriptors within successively wider regions.

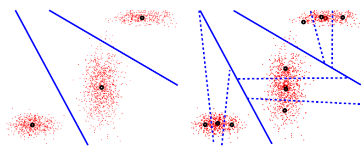
From K. Grauman, B. Leibe

## Pyramid partitions modes

- Two approaches to data-dependent pyramid partitions of feature spaces:
  - Place a multi-dimensional, multi-resolution grid over point sets
  - Create pyramid bins from clusters in the feature space as formed for the creation of the feature vocabulary

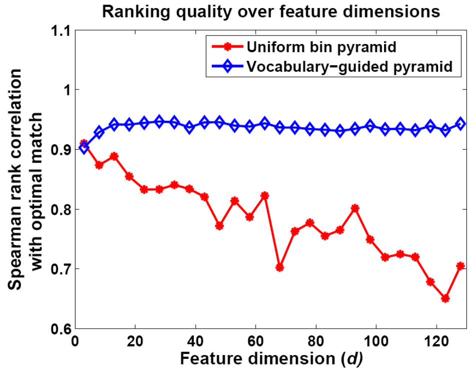


Uniform pyramid bins



Vocabulary-guided pyramid bins

- Vocabulary-guided pyramid match tunes pyramid partitions to the feature distribution. Requires initial corpus of features to determine pyramid structure. Small cost increase over uniform bins: *kL* distances against bin centers to insert points
- It is more accurate than uniform binning for *d* > 100



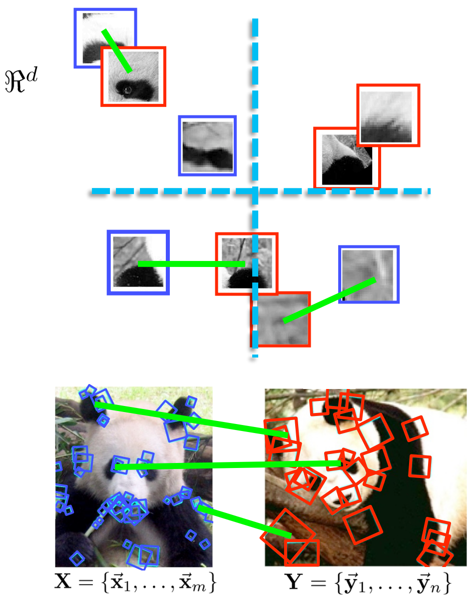

Spearman coefficient is a measure of statistical dependence of two variables:

$$\rho_s = \frac{\sum_i (r_i - \overline{r})(s_i - \overline{s})}{\sqrt{\sum_i (r_i - \overline{r})^2} \sqrt{\sum_i (s_i - \overline{s})^2}}$$

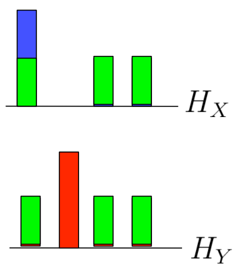
ETH-80 images, sets of SIFT features

Pictures from K. Grauman, B. Leibe

## Histogram intersection at each pyramid level




Histogram intersection counts number of possible matches at a given partitioning.




$$\mathcal{I}(H_X, H_Y) = \sum_j \min\left(H_X(j), H_Y(j)\right)$$
$$= 3$$

From K. Grauman, B. Leibe

## Pyramid match kernel

- Pyramid match kernel:
  - Create multi-resolution feature pyramids: histogram at level *i* has bins of size $2^i$
  - Consider points matched at finest resolution where they fall into same grid cell.
  - Approximate similarity between matched points with worst case similarity at each level.
- Approximate partial match similarity:

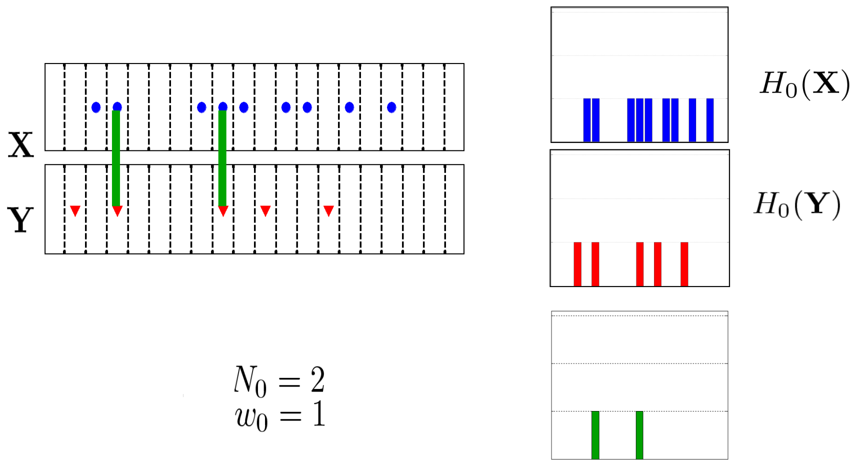
Difference in histogram intersections across levels counts the *number of new pairs* matched at pyramid level *i*

$$K_\Delta = \sum_{i=0}^{L} w_i N_i = \sum_{i=0}^{L} \frac{1}{2^i}\Big(\underbrace{\mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right)}_{\text{matches at level } i} - \underbrace{\mathcal{I}(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y}))}_{\text{matches at previous level}}\Big)$$

Weights $w_i$ of pyramid level *i* are inversely proportional to bin size:
- measure of difficulty of a feature match at level *l*
- normalize kernel values to avoid favoring large sets

## Example of pyramid matching

Level 0

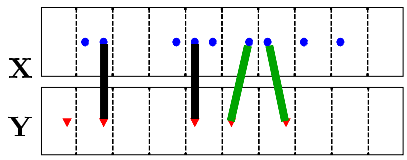


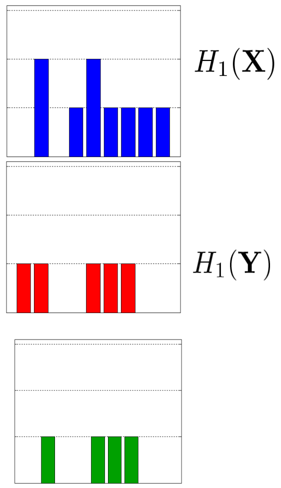

$N_0 = 2$
$w_0 = 1$

$\mathcal{I}_0 = 2$

Pictures from K. Grauman, B. Leibe

Level 1



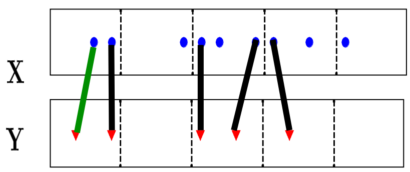



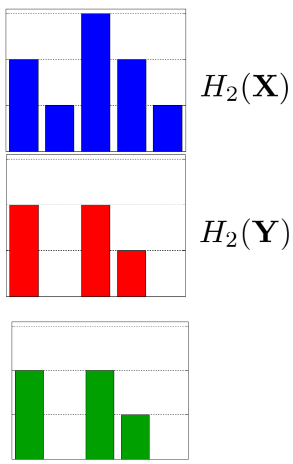

$$N_1 = 4 - 2 = 2$$
$$w_1 = \frac{1}{2}$$

$\mathcal{I}_1 = 4$

Pictures from K. Grauman, B. Leibe

Level 2



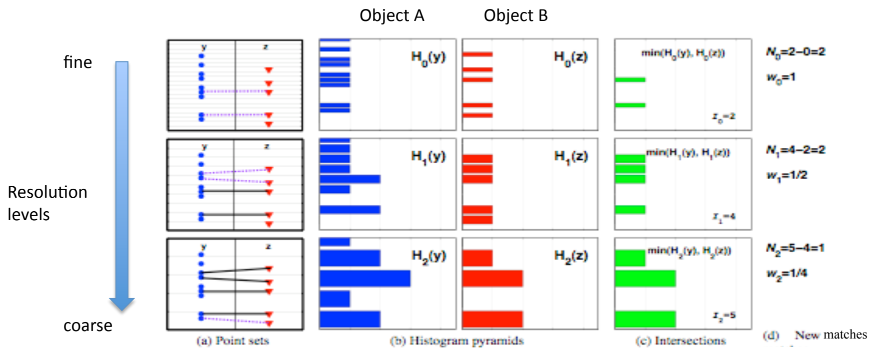




$$N_2 = 5 - 4 = 1$$
$$w_2 = \frac{1}{4}$$

$\mathcal{I}_2 = 5$

Pictures from K. Grauman, B. Leibe

## Pyramid matching



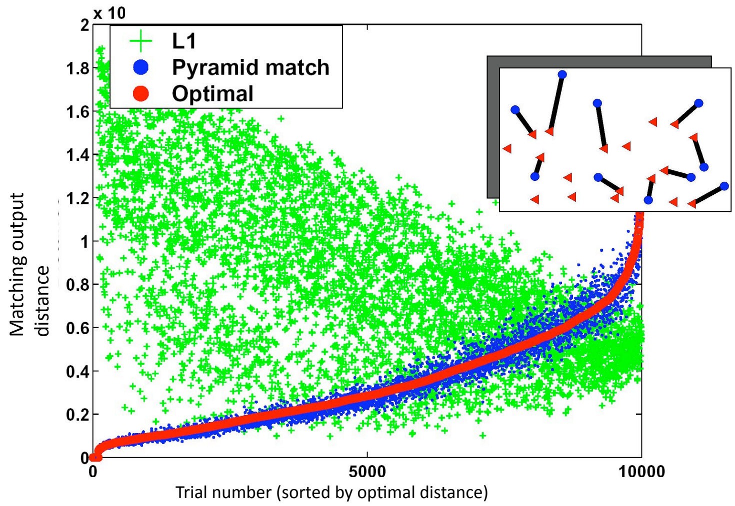

Similarity matching score $$K_{\Delta} = \sum_{i=0}^{L} w_i N_i = 1(2) + \tfrac{1}{2}(2) + \tfrac{1}{4}(1) = 3.25$$

Pictures from K. Grauman, B. Leibe

## Performance figures

- The optimal matching is expensive relative to number of features per image.
  Pyramid matching allows strong computational savings. For sets with *m* features of dimension *d*, and pyramids with *L* levels, computational complexity is linear time complexity:

  - Optimal partial match (Hungarian algorithm) : O($dm^3$)
  - Pyramid partial match: O(dmL)

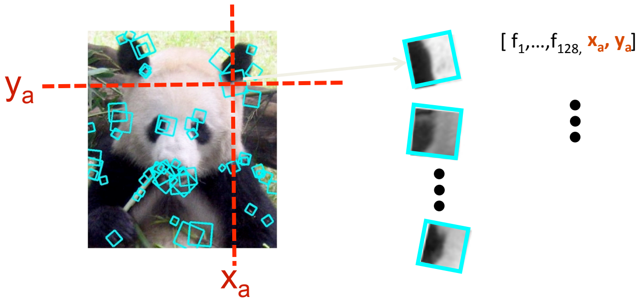



100 sets with 2D points, cardinalities vary between 5 and 100

From Indyk & Thaper

## Matching and local geometry

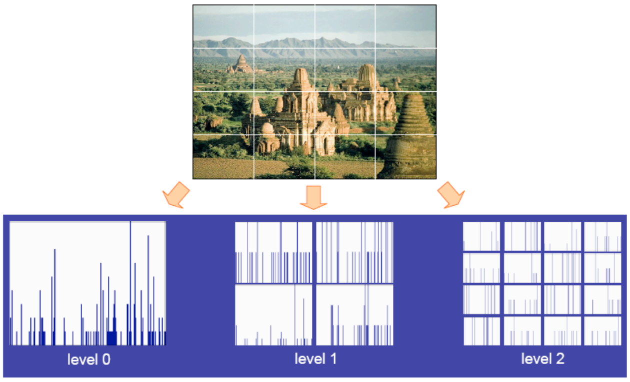
- To consider relative geometry between features a simple solution is to expand feature vectors to include spatial information *before* matching. This forces the matched features both to look the same and have the same spatial layout




Pictures from K. Grauman, B. Leibe

## Spatial Pyramid matching

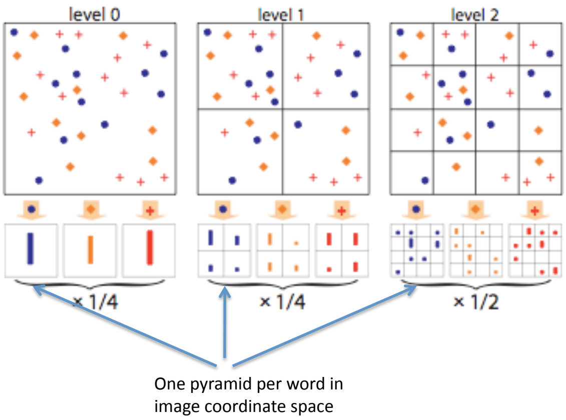
- Spatial pyramid representation can also used to account for spatial information. In this approach feature space is quantized into visual words and hence the Pyramid Match Kernel is computed per visual word. Pyramids are built on the space of image coordinates.
- With spatial pyramid matching features at higher levels are weighted more highly to reflect the fact that higher levels localize the features more precisely.



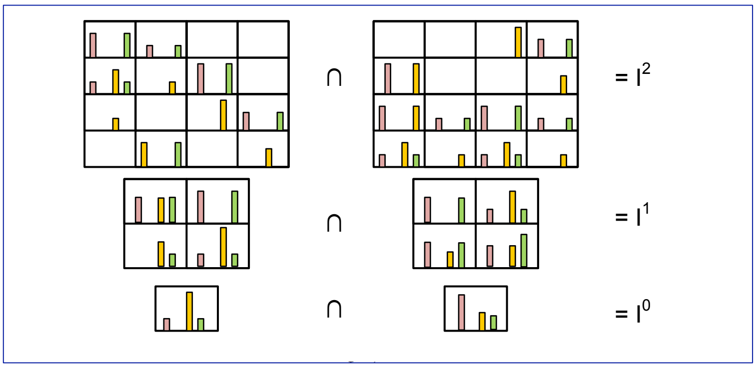

From Lazebnik, Schmid & Ponce, CVPR 2006

## Spatial pyramid match kernel

- Given a set of features $X$, expand the feature notation including the descriptor image position $x_i\, y_i$ and the word index $w_i$
  $X = \{(f_1, x_1 y_1, w_1), (f_2, x_2 y_2, w_2), \ldots\ldots(f_m, x_m y_m, w_m)\}$
- Build one pyramid match *per word* in image coordinate space. Each bin of the histogram pyramid for the j-th word counts how many times that word occurs in the set within the spatial boundaries



- The spatial pyramid matching kernel computes the sum over all the scores of the words pyramid match
  - Level 0 equals standard Bag-of-Words
  - At each level spatial configuration detail importance is increased.
  - Matches are only counted once




$$K_{SPMK} = \sum_{j=1}^{M} k_{PMK}(w_j)$$

Spatial pyramid match

$$k_{PMK}(w_j) = I^L + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}} (I^l - I^{l+1})$$

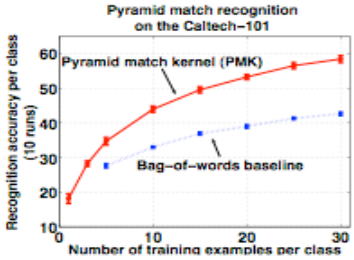
Words pyramid match

## Pyramid Match Kernel for SVM classification

- The Pyramid Match Kernel is positive-definite Mercer kernel and can be used with SVM classifiers to perform partial matching of sets of features:
  - Train 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

- Its convergence is guaranteed and has shown to be robust to clutter, segmentation errors, occlusion…

- The pyramid matching outperforms the bag-of-words approach for object category recognition.




| Kernel | Recognition rate |
|---|---|
| Match *[Wallraven et al.]* | 84% |
| Bhattacharyya affinity *[Kondor & Jebara]* | 85% |
| Pyramid match | 84% |

ETH-80 database 8 object classes
Features: Harris detector, PCA-SIFT descr, *d*=10

From Eichhorn and Chapelle 2004 Slide from K. Grauman