Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2006-020 March 18, 2006

Pyramid Match Kernels: Discriminative Classification with Sets of Image Features (version 2)

Kristen Grauman and Trevor Darrell

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Pyramid Match Kernels: Discriminative Classification with Sets of Image Features (version 2)*

Kristen Grauman and Trevor Darrell
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
Cambridge, MA, USA

Abstract

Discriminative learning is challenging when examples are sets of features, and the sets vary in cardinality and lack any sort of meaningful ordering. Kernel-based classification methods can learn complex decision boundaries, but a kernel over unordered set inputs must somehow solve for correspondences – generally a computationally expensive task that becomes impractical for large set sizes. We present a new fast kernel function which maps unordered feature sets to multi-resolution histograms and computes a weighted histogram intersection in this space. This "pyramid match" computation is linear in the number of features, and it implicitly finds correspondences based on the finest resolution histogram cell where a matched pair first appears. Since the kernel does not penalize the presence of extra features, it is robust to clutter. We show the kernel function is positive-definite, making it valid for use in learning algorithms whose optimal solutions are guaranteed only for Mercer kernels. We demonstrate our algorithm on object recognition tasks and show it to be accurate and dramatically faster than current approaches.

1. Introduction

A variety of representations used in computer vision consist of unordered sets of features or parts, where each set varies in cardinality, and the correspondence between the features across each set is unknown. For instance, an image may be described by a set of detected local affine-invariant regions, a shape may be described by a set of local descriptors defined at each edge point, or a person's face may be represented by a set of views under different conditions. In such cases, one set of feature vectors denotes a single instance of a particular class of interest (an object, scene, shape, face, etc.), and it is expected that the number of features will vary across examples due to viewpoint changes, occlusions, or inconsistent detections by the interest operator.

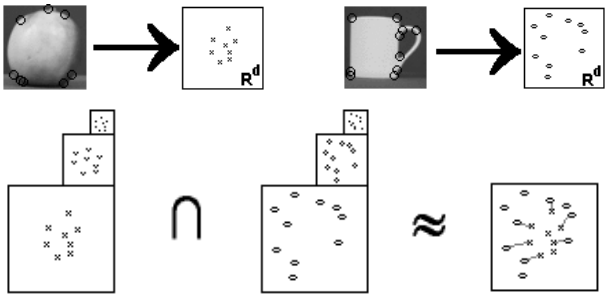


Figure 1: The pyramid match kernel intersects histogram pyramids formed over local features, approximating the optimal correspondences between the sets' features.

To perform learning tasks like categorization or recognition with such representations is challenging. While generative methods have had some success, kernel-based discriminative methods are known to represent complex decision boundaries very efficiently and generalize well to unseen data [29, 26]. For example, the Support Vector Machine (SVM) is a widely used approach to discriminative classification that finds the optimal separating hyperplane between two classes. Kernel functions, which measure similarity between inputs, introduce non-linearities to the decision functions; the kernel non-linearly maps two examples from the input space to the inner product in some feature space. However, conventional kernel-based algorithms are designed to operate on fixed-length vector inputs, where each vector entry corresponds to a particular global attribute for that instance; the commonly used general-purpose kernels defined on $\Re^n$ inputs (e.g., Gaussian RBF, polynomial) are not applicable in the space of vector sets.

In this work we propose a *pyramid match kernel* – a new kernel function over unordered feature sets that allows them to be used effectively and efficiently in kernel-based learning methods. Each feature set is mapped to a multi-resolution histogram that preserves the individual features'

*This technical report serves to update the one previously filed under report number MIT-CSAIL-TR-2005-017.

distinctness at the finest level. The histogram pyramids are then compared using a weighted histogram intersection computation, which we show defines an implicit correspondence based on the finest resolution histogram cell where a matched pair first appears (see Figure 1).

The similarity measured by the pyramid match approximates the similarity measured by the optimal correspondences between feature sets of unequal cardinality (i.e., the *partial matching* that optimally maps points in the lower cardinality set to some subset of the points in the larger set, such that the summed similarities between matched points is maximal). Our kernel is extremely efficient and can be computed in time that is linear in the sets' cardinality. We show that our kernel function is positive-definite, meaning that it is appropriate to use with learning methods that guarantee convergence to a unique optimum only for positive-definite kernels (e.g., SVMs).

Because it does not penalize the presence of superfluous data points, the proposed kernel is robust to clutter. As we will show, this translates into the ability to handle unsegmented images with varying backgrounds or occlusions. The kernel also respects the co-occurrence relations inherent in the input sets: rather than matching features in a set individually, ignoring potential dependencies conveyed by features within one set, our similarity measure captures the features' joint statistics.

Other approaches to this problem have recently been proposed [30, 18, 4, 16, 32, 20, 25], but unfortunately each of these techniques suffers from some number of the following drawbacks: computational complexities that make large feature set sizes infeasible; limitations to parametric distributions which may not adequately describe the data; kernels that are not positive-definite (do not guarantee unique solutions for an SVM); limitations to sets of equal size; and failure to account for dependencies within feature sets.

Our method addresses all of these issues, resulting in a kernel appropriate for comparing unordered, variable-length feature sets within any existing kernel-based learning paradigm. We demonstrate our algorithm with object recognition tasks and show that its accuracy is comparable to current approaches, while requiring significantly less computation time.

2. Related Work

In this section, we review related work on discriminative classification with sets of features, using kernels and SVMs for recognition, and multi-resolution image representations.

Object recognition is a challenging problem that requires strong generalization ability from a classifier in order to cope with the broad variety in illumination, viewpoint, occlusions, clutter, intra-class appearances, and deformations that images of the same object or object class will exhibit. While researchers have shown promising results applying SVMs to object recognition, they have generally used global image features – ordered features of equal length measured from the image as a whole, such as color or grayscale histograms or vectors of raw pixel data [6, 22, 21]. Such global representations are known to be sensitive to real-world imaging conditions, such as occlusions, pose changes, or image noise.

Method	Complexity	C	P	M	U
Match [30]	$O(dm^2)$			x	x
Exponent [18]	$O(dm^2)$		x	x	x
Greedy [4]	$O(dm^2)$	x		x	x
Princ. ang. [32]	$O(dm^3)$	x	x		
Bhattach.'s [16]	$O(dm^3)$	x	x		x
KL-div. [20]	$O(dm^2)$	x			x
Pyramid	$O(dm\ \log D)$	x	x	x	x

Table 1: Comparing kernel approaches to matching unordered sets. Columns show each method's computational cost and whether its kernel captures co-occurrences (C), is positive-definite (P), does not assume a parametric model (M), and can handle sets of unequal cardinality (U). d is vector dimension, m is maximum set cardinality, and D is the feature value range. "Pyramid" refers to the proposed kernel.

Recent work has shown that local features invariant to common image transformations (e.g., SIFT [17]) are a powerful representation for recognition, because the features can be reliably detected and matched across instances of the same object or scene under different viewpoints, poses, or lighting conditions. Most approaches, however, perform recognition with local feature representations using nearest-neighbor (e.g., [1, 10, 27, 3]) or voting-based classifiers followed by an alignment step (e.g., [17, 19]); both may be impractical for large training sets, since their classification times increase with the number of training examples. An SVM, on the other hand, identifies a sparse subset of the training examples (the support vectors) to delineate a decision boundary.

Kernel-based learning algorithms, which include SVMs, kernel PCA, or Gaussian Processes, have become well-established tools that are useful in a variety of contexts, including discriminative classification, regression, density estimation, and clustering [26]. More recently, attention has been focused on developing specialized kernels that can more fully leverage these tools for situations where the data cannot be naturally represented by a Euclidean vector space, such as graphs, strings, or trees.

Several researchers have designed similarity measures that operate on sets of unordered features. See Table 1 for a concise comparison of the approaches. The authors of [30] propose a kernel that averages over the similarities of the best matching feature found for each feature member within the other set. The use of the "max" operator in this

kernel makes it non-Mercer (i.e., not positive-definite – see Section 3), and thus it lacks convergence guarantees when used in an SVM. A similar kernel is given in [18], which also considers all possible feature matchings but raises the similarity between each pair of features to a given power. Both [30] and [18] have a computational complexity that is quadratic in the number of features. Furthermore, both match each feature in a set independently, ignoring potentially useful co-occurrence information. In contrast, our kernel captures the joint statistics of co-occurring features by matching them concurrently as a set.

The method given in [4] is based on finding a sub-optimal matching between two sets using a greedy heuristic; although this results in a non-Mercer kernel, the authors provide a means of tuning the kernel hyperparameter so as to limit the probability that a given kernel matrix is not positive-definite. The authors of [32] measure similarity in terms of the principal angle between the two linear subspaces spanned by two sets' vector elements. This kernel is only positive-definite for sets of equal cardinality, and its complexity is cubic in the number of features. In [25], an algebraic kernel is used to combine similarities given by vector-based kernels, with the weighting chosen to reflect whether the features are in alignment (ordered). When set cardinalities vary, inputs are padded with zeros so as to form equally-sized matrices.

In [16], a Gaussian is fit to each set of vectors, and then the kernel value between two sets is the Bhattacharyya affinity between their Gaussian distributions. As noted by the authors, the method is constrained to using a Gaussian model in order to have a closed form solution. In practice, the method in [16] is also limited to sets with small cardinality, because its complexity is cubic in the number of features. Similarly, the authors of [20] fit a Gaussian to a feature set, and then compare sets using KL-divergence as a distance measure. Unlike the kernels of [16] and [20], which are based on parametric models that assume inputs will fit a certain form, our method is model-free and maintains the distinct data points in the representation.

An alternative approach when dealing with unordered set data is to designate prototypical examples from each class, and then represent examples in terms of their distances to each prototype; standard algorithms that handle vectors in a Euclidean space are then applicable. The authors of [33] build such a classifier for handwritten digits, and use the shape context distance of [1] as the measure of similarity. The issues faced by such a prototype-based method are determining which examples should serve as prototypes, choosing how many there should be, and updating the prototypes properly when new types of data are encountered.

Our feature representation is based on a multi-resolution histogram, or "pyramid", which is computed by binning data points into discrete regions of increasingly larger size. Single-level histograms have been used in various visual recognition systems, one of the first being that of [28], where the intersection of global color histograms was used to compare images. Pyramids have been shown to be a useful representation in a wide variety of image processing tasks – see [12] for a summary.

In [14], multi-resolution histograms are compared with L_1 distance to approximate a least-cost matching of equal-mass global color histograms for nearest neighbor image retrievals. This work inspired our use of a similar representation for point sets. However, unlike [14], our method builds a discriminative classifier, and it compares histograms with a weighted intersection rather than L_1. Our method allows inputs to have unequal cardinalities and thus enables partial matchings, which is important in practice for handling clutter and unsegmented images.

We believe ours is the first work to advocate for the use of a histogram pyramid as an explicit discriminative feature formed over sets, and the first to show its connection to optimal partial matching when used with a hierarchical weighted histogram intersection similarity measure.

3. Approach

Kernel-based learning algorithms [26, 29] are founded on the idea of embedding data into a Euclidean space, and then seeking linear relations among the embedded data. For example, an SVM finds the optimal separating hyperplane between two classes in an embedded space (also referred to as the feature space). A kernel function $K : X \times X \rightarrow \Re$ serves to map pairs of data points in an input space X to their inner product in the embedding space F, thereby evaluating the similarities between all points and determining their relative positions. Linear relations are sought in the embedded space, but a decision boundary may still be non-linear in the input space, depending on the choice of a feature mapping function $\Phi : X \rightarrow F$.

The main contribution of this work is a new kernel function based on implicit correspondences that enables discriminative classification for unordered, variable-length sets of vectors. The kernel is provably positive-definite. The main advantages of our algorithm are its efficiency, its use of implicit correspondences that respect the joint statistics of co-occurring features, and its resistance to clutter or "superfluous" data points.

The basic idea of our method is to map sets of features to multi-resolution histograms, and then compare the histograms with a weighted histogram intersection measure in order to approximate the similarity of the best partial matching between the feature sets. We call the proposed kernel a "pyramid match kernel" because input sets are converted to multi-resolution histograms.

3.1. The Pyramid Match Kernel

We consider an input space X of sets of d-dimensional feature vectors whose values in each dimension have a maximal range D and whose minimum inter-vector distance is 1:[1]

$$X = \left\{ \mathbf{x} | \mathbf{x} = \left\{ [f_1^1, \ldots, f_d^1], \ldots, [f_1^{m_\mathbf{x}}, \ldots, f_d^{m_\mathbf{x}}] \right\} \right\}, \quad (1)$$

where $m_\mathbf{x}$ varies across instances in X.

The feature extraction function Ψ is defined as:

$$\Psi(\mathbf{x}) = [H_{-1}(\mathbf{x}), H_0(\mathbf{x}), \ldots, H_L(\mathbf{x})], \quad (2)$$

where $L = \lceil \log_2 D \rceil$, $\mathbf{x} \in X$, $H_i(\mathbf{x})$ is a histogram vector formed over data $\mathbf{x}$ using d-dimensional bins of side length 2^i, and $H_i(\mathbf{x})$ has a dimension $r_i = \left(\frac{D}{2^i}\right)^d$. In other words, $\Psi(\mathbf{x})$ is a vector of concatenated histograms, where each subsequent component histogram has bins that double in size (in all d dimensions) compared to the previous one. The bins in the finest-level histogram H_0 are small enough that each unique d-dimensional data point from sets in X falls into its own bin, and then the bin size increases until all data points from sets in X fall into a single bin at level L.

The pyramid match kernel K_Δ measures similarity between point sets based on implicit correspondences found within this multi-resolution histogram space. The similarity between two input sets is defined as the weighted sum of the number of feature matchings found at each level of the pyramid formed by Ψ:

$$K_\Delta\left(\Psi(\mathbf{y}), \Psi(\mathbf{z})\right) = \sum_{i=0}^{L} w_i N_i, \quad (3)$$

where N_i signifies the number of newly matched pairs at level i. A new match is defined as a pair of features that were not in correspondence at any finer resolution level.

The kernel implicitly finds correspondences between point sets, if we consider two points matched once they fall into the same histogram bin (starting at the finest resolution level where each point is guaranteed to be in its own bin). The matching is equivalent to a hierarchical process: vectors not found to correspond at a high resolution have the opportunity to be matched at lower resolutions. For example, in Figure 2, there are two points matched at the finest scale, two new matches at the medium scale, and one at the coarsest scale. K_Δ's output value reflects the overall similarity of the matching: each newly matched pair at level i contributes a value w_i that is proportional to how similar two points matching at that level must be, as determined by the bin size. Note that the sum in Eqn. 3 starts with index $i = 0$, because the definition of Ψ insures that no points match at level $i = -1$.

[1] This may be enforced by scaling the data to integer values.

To calculate N_i, the kernel makes use of a histogram intersection function $\mathcal{I}$, which measures the "overlap" between two histograms' bins:

$$\mathcal{I}\left(\mathbf{A}, \mathbf{B}\right) = \sum_{j=1}^{r} \min\left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}\right), \quad (4)$$

where $\mathbf{A}$ and $\mathbf{B}$ are histograms with r bins, and $\mathbf{A}^{(j)}$ denotes the count of the j^{th} bin of $\mathbf{A}$.

Histogram intersection effectively counts the number of points in two sets which match at a given quantization level, i.e., fall into the same bin. To calculate the number of newly matched pairs N_i induced at level i, it is sufficient to compute the difference between successive histogram levels' intersections:

$$N_i = \mathcal{I}\left(H_i(\mathbf{y}), H_i(\mathbf{z})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{y}), H_{i-1}(\mathbf{z})\right), \quad (5)$$

where H_i refers to the i^{th} component histogram generated by Ψ in Eqn. 2. Note that the kernel is not searching explicitly for similar points – it never computes distances between the vectors in each set. Instead, it simply uses the change in intersection values at each histogram level to count the matches as they occur.

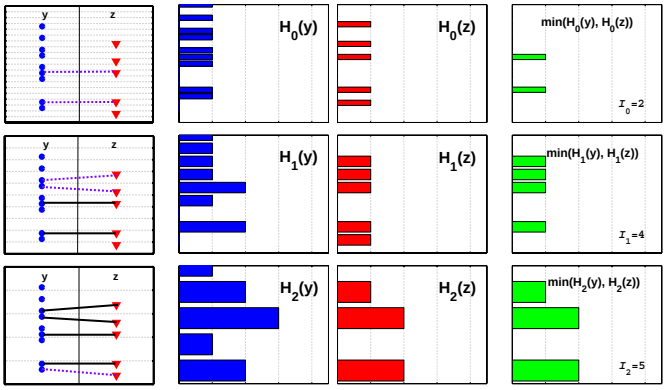
The number of new matches found at each level in the pyramid is weighted according to the size of that histogram's bins: matches made within larger bins are weighted less than those found in smaller bins. Since the largest diagonal of a d-dimensional hypercube bin with sides of length 2^i has length $2^i\sqrt{d}$, the maximal distance between any two points in one bin doubles at each increasingly coarser histogram in the pyramid. Thus, the number of new matches induced at level i is weighted by $\frac{1}{2^i}$ to reflect the (worst-case) similarity of points matched at that level. Intuitively, this means that similarity between vectors (features in $\mathbf{y}$ and $\mathbf{z}$)) at a finer resolution – where features are most distinct – is rewarded more heavily than similarity between vectors at a coarser level.

From Eqns. 3, 4, and 5, we define the (un-normalized) pyramid match kernel function:

$$\tilde{K}_\Delta\left(\Psi(\mathbf{y}), \Psi(\mathbf{z})\right) = \sum_{i=0}^{L} \frac{1}{2^i}\left(\mathcal{I}\left(H_i(\mathbf{y}), H_i(\mathbf{z})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{y}), H_{i-1}(\mathbf{z})\right)\right), \quad (6)$$

where $\mathbf{y}, \mathbf{z} \in X$, and $H_i(\mathbf{x})$ is the i^{th} histogram in $\Psi(\mathbf{x})$.

To provide a partial matching score that does not favor large input sets and has no penalty whatsoever for the unmatched features, we normalize this value by the cardinality of the smaller of the two input sets, arriving at the final kernel value $K_\Delta\left(\Psi(\mathbf{y}), \Psi(\mathbf{z})\right) = \frac{\tilde{K}_\Delta(\Psi(\mathbf{y}), \Psi(\mathbf{z}))}{\min(|\mathbf{y}|, |\mathbf{z}|)}$. Alternatively, both sets' sizes may be taken into account by normalizing by the product of each input's self-similarity, for a final kernel value $K_\Delta\left(\Psi(\mathbf{y}), \Psi(\mathbf{z})\right) = \frac{1}{\sqrt{C}}\tilde{K}_\Delta\left(\Psi(\mathbf{y}), \Psi(\mathbf{z})\right)$, where $C = \tilde{K}_\Delta\left(\Psi(\mathbf{y}), \Psi(\mathbf{y})\right) \times \tilde{K}_\Delta\left(\Psi(\mathbf{z}), \Psi(\mathbf{z})\right)$.



(a) Point sets (b) Histogram pyramids (c) Intersections

Figure 2: A pyramid match determines a partial correspondence by matching points once they fall into the same histogram bin. In this example, two 1-D feature sets are used to form two histogram pyramids. Each row corresponds to a pyramid level. H_{-1} is not pictured here because no matches are formed at the finest level. In (a), the set $\mathbf{y}$ is on the left side, and the set $\mathbf{z}$ is on the right. (Points are distributed along the vertical axis, and these same points are repeated at each level.) Light dotted lines are bin boundaries, bold dashed lines indicate a pair matched at this level, and bold solid lines indicate a match already formed at a finer resolution level. In (b) multi-resolution histograms are shown, with bin counts along the horizontal axis. In (c) the intersection pyramid between the histograms in (b) are shown. K_Δ uses this to measure how many new matches occurred at each level. $\mathcal{I}_i$ refers to $\mathcal{I}(H_i(\mathbf{y}), H_i(\mathbf{z}))$. Here, $\mathcal{I}_i = 2, 4, 5$ across levels, and therefore the number of new matches found at each level are $N_i = 2, 2, 1$. The sum over N_i, weighted by $w_i = 1, \frac{1}{2}, \frac{1}{4}$, gives the pyramid match similarity.

In order to alleviate quantization effects that may arise due to the discrete histogram bins, we can combine the kernel values resulting from multiple (T) pyramid matches formed under different multi-resolution histograms with randomly shifted bins. Each dimension of each of the T pyramids is shifted by an amount chosen uniformly at random between 0 and D. This yields T feature mappings $\Psi_1, \ldots, \Psi_T$ that are applied as in Eqn. 2 to map an input set $\mathbf{y}$ to T multi-resolution histograms: $[\Psi_1(\mathbf{y}), \ldots, \Psi_T(\mathbf{y})]$. For inputs $\mathbf{y}$ and $\mathbf{z}$, the combined kernel value is then $\sum_{j=1}^{T} K_\Delta\left(\Psi_j(\mathbf{y}), \Psi_j(\mathbf{z})\right)$.

3.2. Partial Match Correspondences

Our kernel allows sets of unequal cardinalities, and therefore it enables *partial matchings*, where the points of the smaller set are mapped to some subset of the points in the larger set. Dissimilarity is only judged on the most similar part of the empirical distributions, and superfluous data points are ignored; the result is a robust similarity measure that accommodates inputs expected to contain extraneous vector entries. This is a common situation when recognizing objects in images, due for instance to background variations, clutter, or changes in object pose that cause different subsets of features to be visible. Thus, the proposed kernel is equipped to handle unsegmented examples, as we will demonstrate in Section 4.

By construction, the pyramid match offers an approximation of the optimal correspondence-based matching between two feature sets, in which the overall similarity between corresponding points is maximized. When input sets have equal cardinalities, histogram intersection can be reduced to an L_1 distance: $\mathcal{I}(H(\mathbf{y}), H(\mathbf{z})) = m - \frac{1}{2}||H(\mathbf{y}) - H(\mathbf{z})||_{L_1}$ if $m = |\mathbf{y}| = |\mathbf{z}|$ [28]. Intersection over the pyramid with weights set to $w_i = \frac{1}{2^i}$ then strictly approximates the optimal bipartite matching [14]. With variable cardinalities no similar proof is available, but we show empirically below that the intersection of multi-resolution histograms approximates the best partial matching both in simulation and in practice.

Since the pyramid match defines correspondences across entire sets simultaneously, it inherently accounts for dependencies between various features occurring in one set. In contrast, previous approaches have used each feature in a set to independently index into the second set; this ignores possibly useful information that is inherent in the co-occurrence of a set of distinctive features, and it 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 [30, 18].

3.3. Satisfying Mercer's Condition

Only positive semi-definite kernels guarantee an optimal solution to kernel-based algorithms based on convex optimization, including SVMs. According to Mercer's theorem, a kernel K is positive semi-definite if and only if

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle, \ \forall \mathbf{x}_i, \mathbf{x}_j \in X, \qquad (7)$$

where $\langle \cdot \rangle$ denotes a scalar dot product. This insures that the kernel corresponds to an inner product in some feature space, where kernel methods can search for linear relations [26].

Histogram intersection on single resolution histograms over multi-dimensional data is a positive-definite similarity function [21]. Using this construct and the closure properties of valid kernel functions, we can show that the pyramid match kernel is a Mercer kernel. The definition given in Eqn. 6 is algebraically equivalent to

$$K_\Delta\left(\Psi(\mathbf{y}), \Psi(\mathbf{z})\right) = \frac{\min(|\mathbf{y}|, |\mathbf{z}|)}{2^L} + \sum_{i=0}^{L-1} \frac{1}{2^{i+1}} \mathcal{I}\left(H_i(\mathbf{y}), H_i(\mathbf{z})\right), \quad (8)$$

since $\mathcal{I}\left(H_{-1}(\mathbf{y}), H_{-1}(\mathbf{z})\right) = 0$, and $\mathcal{I}\left(H_L(\mathbf{y}), H_L(\mathbf{z})\right) = \min(|\mathbf{y}|, |\mathbf{z}|)$ by the construction of the pyramid. Given that

Mercer kernels are closed under both addition and scaling by a positive constant [26], we only need to show that the minimum cardinality between two sets ($\min(|\mathbf{y}|, |\mathbf{z}|)$) corresponds to a positive semi-definite kernel.

The cardinality of an input set $\mathbf{x}$ can be encoded as a binary vector containing $|\mathbf{x}|$ ones followed by $Z - |\mathbf{x}|$ zeros, where Z is the maximum cardinality of any set. The inner product between two such expansions is equivalent to the cardinality of the smaller set, thus satisfying Mercer's condition. Note that this binary expansion and the one in [21] only serve to prove positive-definiteness and are never computed explicitly. Therefore, K_Δ is valid for use in existing learning methods that require Mercer kernels.

3.4. Efficiency

The time required to compute Ψ for an input set with m d-dimensional features is $O(dz \log D)$, where $z = \max(m, k)$ and k is the maximum feature value in a single dimension. (Typically $m > k$.) The bin coordinates corresponding to non-zero histogram entries for each of the $\log D$ quantization levels are computed directly during a scan of the m input vectors; these entries are sorted by the bin indices and the bin counts for all entries with the same index are summed to form one entry. This sorting requires only $O(dm + kd)$ time using the radix-sort algorithm, a linear time sorting algorithm that is applicable to the integer bin indices [7]. The histogram pyramid that results is high-dimensional, but very sparse, with only $O(m \log D)$ non-zero entries that need to be stored.

The complexity of K_Δ is $O(dm \log D)$, since computing the intersection values for histograms that have been sorted by bin index requires time linear in the number of non-zero entries (*not* the number of actual bins). Generating multiple pyramid matches with randomly shifted grids simply scales the complexity by T, the constant number of shifts. All together, the complexity of computing both the pyramids and kernel values is $O(Tdm \log D)$. In contrast, current approaches have polynomial dependence on m, which limits the practicality of large input sizes. See Table 1 for complexity comparisons.

4. Results

In this section we show that in simulation the pyramid match kernel approximates the best partial matching of feature sets, and then we report on object recognition experiments with baseline comparisons to other methods.

4.1. Approximate Partial Matchings

As described in Section 3, the pyramid match approximates the optimal correspondence-based matching between two feature sets. While for the case of equal cardinalities it reduces to an L_1 norm in a space that is known to strictly approximate the optimal bijective matching [14], empirically we find the pyramid kernel approximates the optimal partial matching of unequal cardinality sets.

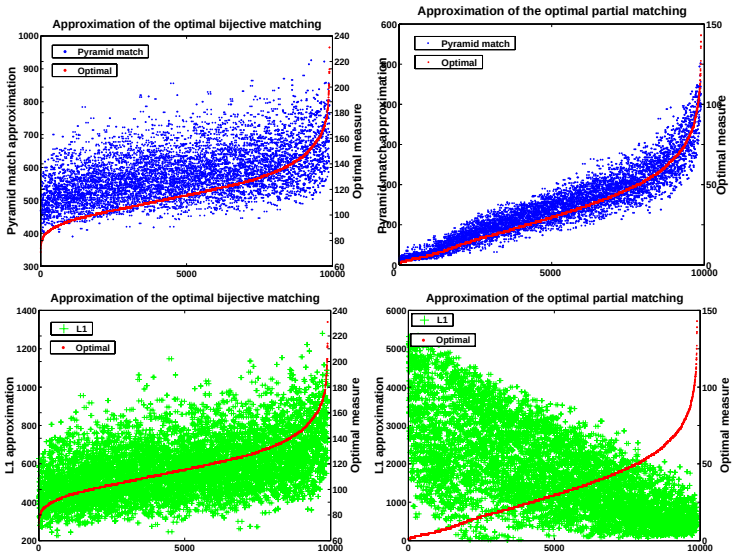


Figure 3: The pyramid match approximates the optimal correspondences, even for sets of unequal cardinalities (right). See text for details. (This figure is best viewed in color.)

We conducted an experiment to evaluate how close the correspondences implicitly assigned by the pyramid match are to the true optimal correspondences – the matching that results in the maximal summed similarity between corresponding points. We compared our kernel's outputs to those produced by the optimal partial matching obtained via a linear programming solution to the transportation problem [23].

We generated two data sets, each with 100 point sets containing 2-D points with values uniformly distributed between one and 1000. In one data set, each point set had equal cardinalities (100 points each), while in the other cardinalities varied randomly from 5 to 100. Figure 3 shows the results of 10,000 pairwise set-to-set comparisons computed according to the correspondences produced by the optimal matching, the pyramid match with $T = 1$, and the L_1 embedding of [14], respectively, for each of these sets. Note that in these figures we plot distance (inverse similarity, with pyramid match weights set to $w_i = 2^i$), and the values were sorted according to the optimal measure's magnitudes for visualization purposes.

This figure shows that our method does indeed find matchings that are consistently on par with the optimal solution. In the equal cardinality case (plots on the left), both the pyramid match and the L_1 embedding produce similar

approximations. However, the pyramid match can also approximate the partial matching for the unequal cardinality case: its matchings continue to follow the optimal matching's trend since it does not penalize outliers. The L_1 embedding fails in this case because it is designed to handle only sets with equal cardinalities and requires all points to match to something. As shown by the plots on the right of Figure 3, the pyramid match maintains a steady approximation factor when sets vary in size, while the L_1 embedding error increases by about a factor of four.

4.2. Object Recognition

For our object recognition experiments we use SVM classifiers, which are trained by specifying the matrix of kernel values between all pairs of training examples. The kernel's similarity values determine the examples' relative positions in an embedded space, and quadratic programming is used to find the optimal separating hyperplane between the two classes in this space. We use the implementation given by [5]. When kernel matrices have dominant diagonals we use the transformation suggested in [31]: a sub-polynomial kernel is applied to the original kernel values, followed by an empirical kernel mapping that embeds the distance measure into a feature space.

Local affine- or scale- invariant feature descriptors extracted from a sparse set of interest points in an image have been shown to be an effective, compact representation (e.g. [17, 19]). This is a good context in which to test our kernel function, since such local features have no inherent ordering, and it is expected that the number of features will vary across examples. In the following we experiment with two publicly available databases and demonstrate that our method achieves comparable object recognition performance at a significantly lower computational cost than other state-of-the-art approaches. All run-times reported below include the time needed to compute both the pyramids and the weighted intersections.

A performance evaluation given in [8] compares the methods of [16, 32, 30] in the context of an object categorization task using images from the publicly available ETH-80 database.[2] The experiment uses eight object classes, with 10 unique objects and five widely separated views of each, for a total of 400 images. A Harris detector is used to find interest points in each image, and various local descriptors (SIFT [17], JET, patches) are used to compose the feature sets. A one-versus-all SVM classifier is trained for each kernel type, and performance is measured via cross-validation, where all five views of an object are held out at once. Note that no instances of a test object are ever present in the training set, so this is a categorization task (as opposed to recognition of the same object).

[2] http://www.vision.ethz.ch/projects/categorization/

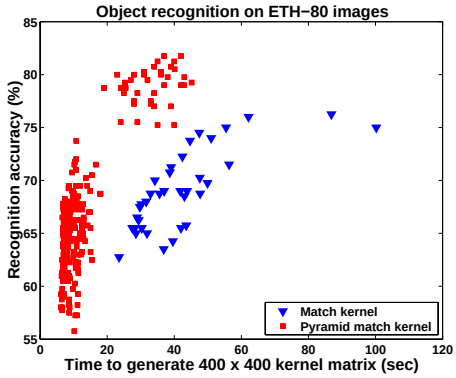


Figure 4: Allowing the same run-time, the pyramid match kernel (with $T = 1$) produces better recognition rates than an approach that computes pairwise distances between features in order to match them. See text for details.

The experiments show the polynomial-time methods of [30] and [16] performing best, with a classification rate of 74% using on average 40 SIFT features per image [8]. Using 120 interest points, the Bhattacharyya kernel [16] achieves 85% accuracy. However, the study also concluded that the cubic complexity of the method given in [16] made it impractical to use the desired number of features.

We evaluated our method on this same subset of the ETH-80 database under the same conditions provided in [8], and it achieved a recognition rate of 83% using PCA-SIFT [15] features from all Harris-detected interest points (averages 153 points per image) and $T = 8$. Restricting ourselves to an average of 40 interest points yields a recognition rate of 73%. Thus our method performs comparably to the others at their best for this data set, but is much more efficient than those tested above, requiring time only linear in the number of features.

In fact, the ability of a kernel to handle large numbers of features can be critical to its success. An interest operator may be tuned to select only the most "salient" features, but in our experiments we found that the various approaches' recognition rates always benefitted from having larger numbers of features per image with which to judge similarity. Figure 4 depicts the run-time versus recognition accuracy of our method as compared to the kernel of [30] (called the "match" kernel), which has $O(dm^2)$ complexity. Each point in the figure represents one experiment; the saliency threshold of the Harris interest operator was adjusted to generate varying numbers of features, thus trading off accuracy versus run-time. Computing a kernel matrix for the same data is significantly faster with the pyramid match kernel, and for similar run-times our method produces much better recognition results.

We also tested our method with a challenging database

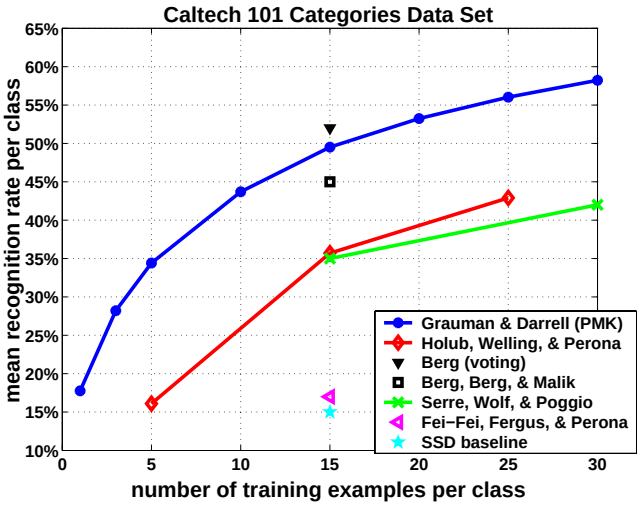


Figure 5: Recognition results on the Caltech101 data set. This plot shows the recognition accuracy using the pyramid match kernel (in blue circles connected by blue lines) for varying numbers of training examples per class, averaged over 10 runs with randomly selected training examples. Results reported by other authors are also shown. All points on the plot refer to recognition rates that have been normalized according to the number of test examples per class. See text for details. (This figure is best viewed in color.)

of 101 object categories recently developed at Caltech [9].[3] This database was obtained using Google Image Search, and many of the images contain a significant amount of intra-class appearance variation.

For this data set, the pyramid match operated on sets of SIFT features projected to 10 dimensions using PCA, with each appearance descriptor concatenated with its corresponding positional feature (image position normalized by image dimensions). The features were extracted on a uniform grid from the images, i.e., no interest operator was applied, and each set had on average 1140 features each. We trained our algorithm with unsegmented images. Since our approach seeks the best correspondence with some subset of the images' features, it explicitly accounts for unsegmented, cluttered data. Classification was again done with a one-vs-all SVM, and we set $T = 3$. We used the current version of the database, which does not contain duplicated images.

Figure 5 shows (in blue) the category recognition results using the pyramid match kernel compared against other published results.[4] The pyramid match scores are given for varying numbers of training set sizes, ranging from one to 30 examples per class. For each training set size, we are displaying the pyramid match accuracy averaged over 10 runs, where for each run we randomly select the training examples and use all the remaining database images as test examples. However, all recognition rates have been normalized according to the number of novel test images per class; that is, the mean recognition rate per class is the average normalized score for all 101 categories.

A standard number of training examples per class is 15. For this size of training set, we obtain a recognition performance of 50% averaged over all classes (0.9% standard deviation), where again the number of correct predictions per class has been normalized by the number of examples in that class. This is an improvement over the 45% performance achieved by the correspondence-based method of Berg et al. [3], and close to the (best) 52% performance reported in Berg [2] using a voting approach.

The method of Berg et al. [3] measures similarity between sets of geometric blur descriptors by approximating the optimal low-distortion correspondences via linear programming, and then uses nearest neighbors to classify images. It requires $O(m^2 n \log n)$ time to compare two images for n test features and m model features, which took about 5 seconds per match in practice on this data set [3]. In [2], nearest neighbor classification is also performed, but on the basis of independent feature voting. Processing one image with voting requires $O(Nm^2)$ time in order to compare its features against all of the features in the N training examples. This breaks down into $O(m^2)$ computation time for every pair of images, or about 0.04 seconds per match in practice on this data set when training with 15 examples per class. In comparison, pyramid match comparisons require only $O(mL)$ time, and in practice averaged 0.002 seconds per image match for this data set. Thus the pyramid match demonstrates recognition accuracy that is competitive with the state-of-the-art, but at a computational cost between one and three orders of magnitude lower.

As shown in Figure 5, when using only one training example per class, our method achieves an average recognition rate per class of 18%. (Note that chance performance with any number of training examples would be just 1%.)

5. Conclusions

We have developed a new fast kernel function that is suitable for discriminative classification with unordered sets of local features. Our pyramid match kernel approximates the optimal partial matching by computing a weighted intersection over multi-resolution histograms, and requires time linear in the number of features. The kernel is robust to clutter since it does not penalize the presence of extra features, respects the co-occurrence statistics inherent in the input sets, and is provably positive-definite. We have applied our kernel to SVM-based object recognition tasks, and demon-

[3] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[4] This updates the results in our ICCV 2005 paper [11], in which an accuracy number is only provided for one training set size, and the performance is not normalized per class.

strated recognition performance with accuracy comparable to current methods, but at a much lower computational cost.

Acknowledgments

We would like to thank John Lee for his help running experiments for this paper, and members of the MIT Vision Interface group and Mark Stephenson for reading earlier drafts.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(24):509–522, April 2002.

[2] A. Berg. *Shape Matching and Object Recognition*. Ph.d. thesis, U.C. Berkeley, Computer Science Division, Berkeley, CA, December 2005.

[3] A. Berg, T. Berg, and J. Malik. Shape Matching and Object Recognition using Low Distortion Correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, June 2005.

[4] S. Boughhorbel, J-P. Tarel, and F. Fleuret. Non-Mercer Kernels for SVM Object Recognition. In *British Machine Vision Conference*, London, UK, Sept 2004.

[5] C. Chang and C. Lin. *LIBSVM: a library for SVMs*, 2001.

[6] O. Chapelle, P. Haffner, and V. Vapnik. SVMs for Histogram-Based Image Classification. *Transactions on Neural Networks*, 10(5), Sept 1999.

[7] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[8] J. Eichhorn and O. Chapelle. Object Categorization with SVM: Kernels for Local Features. Technical report, MPI for Biological Cybernetics, 2004.

[9] L. Fei-Fei, R. Fergus, and P. Perona. Learning Generative Visual Models from Few Training Examples: an Incremental Bayesian Approach Tested on 101 Object Cateories. In *Workshop on Generative Model Based Vision*, Washington, D.C., June 2004.

[10] K. Grauman and T. Darrell. Fast Contour Matching Using Approximate Earth Mover's Distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington D.C., June 2004.

[11] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Beijing, China, October 2005.

[12] E. Hadjidemetriou, M. Grossberg, and S. Nayar. Multiresolution Histograms and their Use for Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(7):831–847, July 2004.

[13] A. Holub, M. Welling, and P. Perona. Exploiting Unlabelled Data for Hybrid Object Classification. In *NIPS 2005 Workshop in Inter-Class Transfer*, Whistler, B.C., December 2005.

[14] P. Indyk and N. Thaper. Fast Image Retrieval via Embeddings. In *3rd Intl Wkshp on Statistical and Computational Theories of Vision*, Nice, France, Oct 2003.

[15] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, D.C., June 2004.

[16] R. Kondor and T. Jebara. A Kernel Between Sets of Vectors. In *Proccedings of International Conference on Machine Learning*, Washington, D.C., Aug 2003.

[17] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, Jan 2004.

[18] S. Lyu. Mercer Kernels for Object Recognition with Local Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, June 2005.

[19] K. Mikolajczyk and C. Schmid. Indexing Based on Scale Invariant Interest Points. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Vancouver, Canada, July 2001.

[20] P. Moreno, P. Ho, and N. Vasconcelos. A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications. In *Neural Information Processing Systems (NIPS)*, Vancouver, Dec 2003.

[21] F. Odone, A. Barla, and A. Verri. Building Kernels from Binary Strings for Image Matching. *IEEE Trans. on Image Processing*, 14(2):169–180, Feb 2005.

[22] D. Roobaert and M. Van Hulle. View-Based 3D Object Recognition with Support Vector Machines. In *IEEE Intl Workshop on Neural Networks for Signal Processing*, Madison, WI, Aug 1999.

[23] Y. Rubner, C. Tomasi, and L. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[24] T. Serre, L. Wolf, and T. Poggio. Object Recognition with Features Inspired by Visual Cortex. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, June 2005.

[25] A. Shashua and T. Hazan. Algebraic Set Kernels with Application to Inference Over Local Image Representations. In *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, Dec 2005.

[26] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[27] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Nice, Oct 2003.

[28] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[29] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998.

[30] C. Wallraven, B. Caputo, and A. Graf. Recognition with Local Features: the Kernel Recipe. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Nice, France, Oct 2003.

[31] J. Weston, B. Scholkopf, E. Eskin, C. Leslie, and W. Noble. Dealing with Large Diagonals in Kernel Matrices. In *Principles of Data Mining and Knowledge Discovery*, volume 243 of *SLNCS*, 2002.

[32] L. Wolf and A. Shashua. Learning Over Sets Using Kernel Principal Angles. *Journal of Machine Learning Research*, 4:913–931, Dec 2003.

[33] H. Zhang and J. Malik. Learning a Discriminative Classifier Using Shape Context Distances. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Madison, WI, June 2003.