Relaxed Matching Kernels For Robust Image Comparison

Andrea Vedaldi Stefano Soatto
University of California at Los Angeles, 90095, USA
{vedaldi,soatto}@cs.ucla.edu
http://vision.ucla.edu/

Abstract

The popular bag-of-features representation for object recognition collects signatures of local image patches and discards spatial information. Some have recently attempted to at least partially overcome this limitation, for instance by "spatial pyramids" and "proximity" kernels. We introduce the general formalism of "relaxed matching kernels" (RMKs) that includes such approaches as special cases, allow us to derive useful general properties of these kernels, and to introduce new ones. As an example, we introduce a kernel based on matching graphs of features and one based on matching information-compressed features. We show that all RMKs are competitive and outperform in several cases recently published state-of-the-art results on standard datasets. However, we also show that a proper implementation of a baseline bag-of-features algorithm can be extremely competitive, and outperform the other methods in some cases.

1. Introduction

Many visual tasks, such as visual recognition, categorization or three-dimensional reconstruction, hinge on establishing at least partial correspondence between different images affected by intrinsic variability of the underlying scene (e.g. "chair"), as well as by variability due to changes in viewpoint and illumination. In order to mitigate the effects of occlusions of line-of-sight, many methods employ a representation of the image in terms of "local features." These are *statistics*, i.e. functions of the image, defined in a neighborhood of a discrete (finite) set of locations, or "keypoints." Such methods, however, vary significantly in how such locations are treated in the representation. At one end of the spectrum are so-called "constellation models" that allow for affine transformations of keypoint locations [24], or more general "deformable templates" that allow for more general transformations, for instance represented by a finite-dimensional thin-plate spline [1]. At the other end of the spectrum are so-called "bags of features" (BoF) methods [2], that discard the location of the features altogether [7].

The fact that BoF methods have been so successful in visual categorization tasks may seem surprising. A possible reason is that, as we showed in [23], achieving viewpoint invariant image representations forces to discard shape information. However, this does not necessarily mean that a fully invariant representation is preferable to one which is perhaps less invariant, but more discriminative. In this context, works such as [11, 13] proposed variants of the bag-of-feature model that tries to capture part of the spatial information as well. In particular, they propose kernels for image comparison which are based on a bag-of-feature representation augmented with spatial information.

In this paper we build upon those works and define a general family of kernels, called "relaxed matching kernels" (RMK) (Sect. 2). This family include as special cases and unifies existing approaches such as the pyramid matching kernel [7], the spatial pyramid matching kernel [11] as well as the proximity distribution kernel [13]. We study interesting properties shared by these kernels and we show that all of them can be computed efficiently. This helps understanding the difference between these approaches, and at least in one case it highlights inconsistencies in the weighting scheme and suggests a better kernel. More importantly, our approach allows us to *define new kernels*, for instance the "graph-matching kernel" (GMK) and agglomerative-information-bottleneck kernel (AIBMK) proposed in Sect. 3.

In Sect. 4.1 we test GMK on matching graphs of generic features, such as those used in the "sketch" [8], for wide-baseline correspondence. We show that, even when features are ambiguous and their identity becomes unstable due to viewpoint changes, the graph matching is robust enough to absorb much of the variability. Finally, in Sect 4.1 we compare various kernels on the task of object recognition on benchmark datasets such as Graz-02 and Pascal-05. We show that our kernels are very competitive with respect to state of the art [21, 13]. We also show, however, that a good baseline implementation of bag-of-features is very competitive with this more advanced methods, an is capable to out-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

perform those and previously published sate-of-the-art results in some cases.

1.1. Bag-of-Features and Beyond

Constructing the Bag-of-Features (BoF) representation [2] of an image starts from the extraction of local image features. First, the image I is decomposed in a number of interest regions. To this end, several operators (feature detectors) are available, ranging from the selection of random patches [16] to the extraction of scale or affine covariant blobs and corners [15]. This results in a list $l_1, \dots, l_N$ of feature locations (and the associated regions). Then the appearance of each region is encoded by a compact but discriminative statistic (feature descriptor). Again, several operators can be used, many of which are based on computing an histogram of the image intensities or gradients [14]. This results in a second list $d_1, \dots, d_N$ of feature descriptors.

The locations $l_1, \dots, l_N$ are then disregarded and the image is represented by the distribution of the feature descriptors $d_1, \dots, d_N$ alone. The distribution is estimated by quantizing the descriptor space $\mathcal{F}$ and then computing an histogram[1] $h(b)$ of the occurrence of the quantized descriptors (it is also possible to avoid quantizing altogether [17]). The quantization $B \subset 2^{\mathcal{F}}$ may be obtained by a variety of methods, such as K-means or regular partitioning [7, 21]. By analogy with the bag-of-words model of text analysis, the quantized descriptors $b_1, \dots, b_N \in B$ are also called *visual words* and the quantization B *visual dictionary*.

Comparing two images I^1 and I^2 is then reduced to evaluating the similarity $K(h^1, h^2)$ of the respective bag-of-features $h^k(b)$, $k = 1, 2$ representations. Recently [25] has shown that the χ^2 Radial Basis Function (RBF) kernel (Sect. 2) yields particularly good performances in object categorization with the advantage of being directly operable in an SVM classifier.

A problem with the dictionary approach to BoF is the choice of the resolution of the visual dictionary B. An excessively fine quantization causes features from two images to never match (overfitting), while an excessively coarse quantization yields non-discriminative histograms (bias). Grauman et al. [7] proposed *Pyramid Matching Kernel* to overcome this issue. The idea is to work with a sequence of R progressively coarser dictionaries $B_0, B_1, \dots, B_{R-1}$ and to define a similarity measure as a positive combination of the BoF similarities at the various levels. The formulation yields a proper Mercer (positive definite) kernel.

While BoF is a powerful paradigm, disregarding completely the image geometry limits the discriminative power of the representation. Several attempts have been made to extend BoF to account for geometric information. The easiest way is to append the interest point locations to the descriptors (Sect. 4 of [7]). Lazebnik et al. [11] extend this idea and introduce the *Spatial Pyramid Matching Kernel* (SPMK): They propose to use quantized pairs (l_i, d_i) of interest point location-descriptor as element of the base visual dictionary B_0. The pyramid $B_0, B_1, \dots, B_{R-1}$ is then formed by coarsening the quantization of the location component l only. In this way, the representation captures the distribution of both the appearance and location of the interest points.

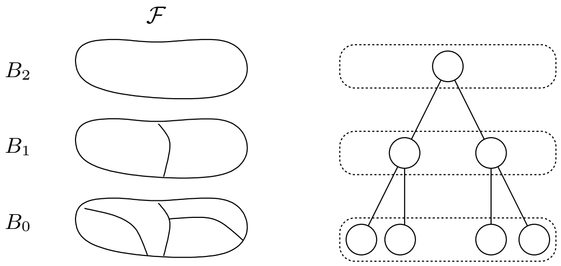


Figure 1: **RMK construction: agglomerative tree**. *Left.* The feature space $\mathcal{F}$ and a sequence of three relaxations B_0, B_1 and B_2. *Right.* The agglomerative tree represents the merging operations transforming a relaxation to the next. Each relaxation B_r corresponds to a cut of the tree (dotted boxes).

A limitation of this approach is that, since the location l is expressed in absolute coordinates, the representation is unsuitable for objects which present large variations in pose. To address this issue, Ling et al. [13] introduced the *Proximity Distribution Kernel* (PDK). The idea is to start from triplets (d_i, d_j, ρ_{ij}), where d_i and d_j are interest points descriptors and ρ_{ij} is their (nearest neighbors) distance. Successive relaxations merge increasing values of the ρ component (Sect. 2). Since ρ is a relative quantity, the limitation of SPMK is removed.

2. Relaxed Matching Kernels

In this section we introduce the "relaxed matching kernels" which generalize PMK, SPMK and PDK.

Construction. Let $B_0 \subset 2^{\mathcal{F}}$ a quantization of the feature space $\mathcal{F}$ (base visual dictionary). To obtain coarser quantizations B_r, we recursively merge bins $b \in B_0$ (Fig 1). The result of this process is an *agglomerative tree*, whose nodes are bins and parents are obtained from children by merging.[2]

The base dictionary B_0 corresponds to the leaves of the agglomerative tree and the coarser dictionaries B_r corre-

[1] We assume that histograms are normalized to one.

[2] In practice the tree might be a forest if one stops merging before all bins are merged into one (but one can always assume that the latter is the case).

spond to tree "cuts". A *cut* (Fig. 1) is just a subset B_r of the tree nodes such that any leaf $b \in B_0$ is descendent of exactly one node $b' \in B_r$ of the cut. Cuts have the property of preserving the mass of the dictionary: If $h_{B_0}(b), b \in B_0$ is an histogram on the finer dictionary B_0, then its projection $h_{B_r}(b)$ on the cut B_r satisfies

$$\sum_{b \in B_0} h_{B_0}(b) = 1 = \sum_{b \in B_r} h_{B_r}(b).$$

We compare images I^k, $k = 1, 2$ by comparing histograms of features defined on corresponding cuts. Given a cut B_r, the similarity measure is given by

$$F_r = k_1(h^1_{B_r}, h^2_{B_r}) = \sum_{b \in B_r} \min\{h^1_{B_r}(b), h^2_{B_r}(b)\} \quad (1)$$

To make the match robust, we adopt a "multiscale" approach. We consider multiple cuts B_r at increasing *relaxation levels* $r = 0, 1 \dots, R-1$ and define

$$K(h^1, h^2) = \sum_{r=0}^{R-1} w_r F_r, \quad (2)$$

where $w_r \geq 0$ is a sequence of weights that establish the relative importance of the relaxations. We define this quantity *relaxed matching kernel* (RMK).

Base kernel, Mercer's condition, and RBF. The RMK is a positive definite (p.d.) kernel [19] since each term $k_1(h^1_{B_r}, h^2_{B_r})$ of the summation (1) is p.d. [9] and the weights w_r are non-negative [19]. Interestingly, Hein et al. [9] provide a whole family of base kernels that can be substituted to the l_1 kernel k_1 in (1). This family includes the χ^2 and Hellinger's kernels

$$k_{\chi^2}(p, q) = 2 \sum_i \frac{p_i q_i}{p_i + q_i}, \quad k_H(p, q) = \sum_i \sqrt{p_i q_i}.$$

All of these choices yield p.d. RMKs (another useful property is that the kernels are normalized to one, i.e. $k(p, p) = 1$).

Finally, each base kernel corresponds to a distance $d^2(p, q)$ by the formula $d^2(p, q) = 2 - 2k(p, q)$. So, for instance, $k_1(p, q)$ corresponds to $d_1^2(p, q) = \|p - q\|_1$ and the χ^2 and Hellinger's kernels correspond to

$$d^2_{\chi^2}(p, q) = \sum_i \frac{(p_i - q_i)^2}{p_i + q_i}, \; d^2_H(p, q) = \sum_i (\sqrt{p}_i - \sqrt{q}_i)^2.$$

These distances can be used to define corresponding RBF kernels by setting $k(p, q) = \exp(-\gamma d^2(p, q))$, where $\gamma > 0$ is a tuning parameter. These kernels are also p.d. [9].

This flexibility in the choice of the base kernel is interesting as, for instance, [25] showed that the χ^2 RBF kernel may perform better than the l_1 kernel (on which PMK, SPMK and PDK are based) for the task of object categorization.

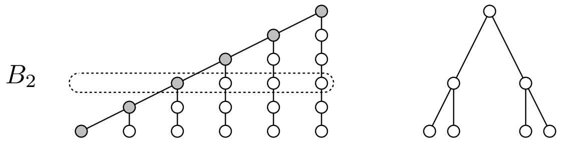


Figure 2: **RMK: agglomerative trees for PDK, PMK and SPMK.** *Left.* PDK relaxations merge successive values of the distance component ρ, yielding a "linear" agglomerative tree. As an illustration, we highlight the cut corresponding to relaxation B_2. PDK fails to be a proper RMK, however, as it considers only the shaded nodes and is not normalized. *Right.* PMK and SPMK relaxations are obtained by merging octaves of the scale space, yielding a "logarithmic" agglomerative tree.

PMK, SPMK, and PDK. The RMK construction encompasses the approaches discussed in Set. 1.1. In PMK the feature space $\mathcal{F}$ is the set of descriptors d, B_0 is a regular partition of $\mathcal{F}$ and B_r are obtained by recursively merging such partitions, reducing by half the resolution of the quantization. The SPMK is similar, except that relaxations operate on the location component l of the features (Sect. 1.1). The corresponding agglomerative tree height is logarithmic in the size of the base dictionary B_0 (Fig. 2).

In PDK B_0 is obtained by quantizing the descriptor component d_i and d_j of the triplets (d_i, d_j, ρ_{ij}) (ρ_{ij} is already discrete). Then the successive relaxations B_r are defined by merging triplets that have distance $\rho_{ij} \leq r + 1$. Still PDK is not a proper RMK because (a) the histograms are not normalized and (b) at each level the comparison (1) is defined as

$$k_{\text{PDK}}(h^1_{B_r}, h^2_{B_r}) = \\ \sum_{d_1, d_2} \min \left\{ \sum_{\rho \leq r+1} h^1_{B_0}(d_1, d_2, \rho), \sum_{\rho \leq r+1} h^2_{B_0}(d_1, d_2, \rho) \right\}$$

and misses part of the mass. Specifically, the RMK version of PDK (Fig. 2) yields

$$k_{\text{PDK/RMK}}(h^1_{B_r}, h^2_{B_r}) = k_{\text{PDK}}(h^1_{B_r}, h^2_{B_r}) \\ + \sum_{d_1, d_2} \sum_{\rho > r+1} \min \left\{ h^1_{B_0}(d_1, d_2, \rho), h^2_{B_0}(d_1, d_2, \rho) \right\}.$$

Meaning of the weights. Define $W_r = \sum_{q=0}^{r} w_q$ and $f_r = F_r - F_{r-1}$, $W_{-1} = F_{-1} = 0$. Then the RMK (2) may be

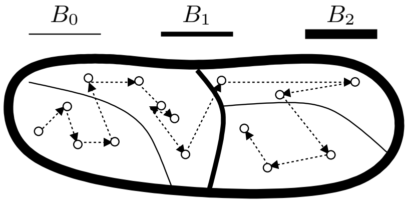


Figure 3: **RMK computation: feature visit order.** The figure shows the feature space $\mathcal{F}$ and the quantization B_0, B_1 and B_2 of Fig. 1. The dots represents the features f_i^k and the dotted arrows a possible visiting order. Notice that the visit traverses all the features of a bin $b \in B_r$ before passing to the successive bin $b' \in B_r$, for all relaxations $r = 0, 1, 2$.

rewritten as

$$K = \sum_{r=0}^{R-1} w_r F_r = \sum_{r=0}^{R-1} (W_{R-1} - W_{r-1}) f_r. \qquad (3)$$

An interesting property of the successive relaxations, proved in Theorem 1, is that F_r is a monotonically increasing quantity (for a large choice of base kernels, including all the popular ones). Moreover, if the last relaxation level corresponds to merging the whole feature space into a single bin, since the base kernel is normalized we also have $F_R = 1$. Therefore we can interpret F_r as a cumulative distribution function and the summation (3) as the expected value $K = E_{f_r}[W_{R-1} - W_{r-1}]$ of the function $W_{R-1} - W_{r-1}$ of the random variable r with (discrete) density f_r. Notice that f_r assigns more mass to the relaxation levels r for which there is an abrupt increase in the matching score F_r. Since $W_{R-1} - W_{r-1}$ decays with increasing relaxation r (the weights are positive), this means that the score is large if the two image statistics match early in the relaxation sequence. In other words, the kernel is looking for the finer relaxation level for which the statistics match well.[3]

For instance the PMK and SPMK kernels have exponentially decaying integral weights of the form $W_r \propto -e^{-\lambda r}$, $\lambda > 0$ (up to a positive factor and offset). In fact, computing the differences $W_r - W_{r-1}$ yields $w_r \propto e^{-\lambda r}$ and we have

$$K_{\text{PMK}} \propto \sum_{r=0}^{R-1} (e^{-\lambda r} - e^{-\lambda R}) f_{r-1} \propto \sum_{r=0}^{R-1} e^{-\lambda r} F_r.$$

For the PDK/RMK kernel we have $w_r = 1$, $W_r = r$ and

$$K_{\text{PDK}} = \sum_{r=0}^{R-1} F_r = \sum_{r=0}^{R-1} (R - r) f_r$$

so the weights are linearly decaying.

Computation. We show next that computing an RMK it is a fast operation as it it is linear in the number of features and relaxation levels.[4]

Let f_i^k, $i = 1, \dots, N_k$, $k = 1, 2$ be the features extracted from images I^1 and I^2 and quantized to the base level B_0. Let F_r, L_r^1, L_r^2, $r = 0, \dots, R-1$ be three accumulators initialized to zero.

First, we show how to calculate F_r according to the definition (1) for a fixed relaxation level r. To do this, we need to compare histograms $h_{B_r}^1$ and $h_{B_r}^2$ defined over bins $B_r = \{b_{r1}, \dots, b_{rM}\}$. We start by visiting all the features f_i^k that belong to the first bin b_{r1}, incrementing the value of the respective accumulators L_r^k. When there are no more features in b_{r1}, we compute $\min\{h_{B_r}^1(b_{r1}), h_{B_r}^2(b_{r2})\} = \min\{L_r^1, L_r^2\}$ as of equation (1), accumulate the result to F_r, set L_r^1 and L_r^2 to zero, and proceed to the next bin b_{r2}. When all bins $b_{rm} \in B_r$ are exhausted, F_r holds the value (1).

This process can be extended to work *simultaneously* for all relaxation levels $r = 0, \dots, R-1$. This is possible because bins b_{ri} at level r are fully contained in bins $b_{r+1,j}$ at level $r+1$, so visiting the features belonging to $b_{r+1,j}$ can be done by visiting the features belonging respectively to all the bins $b_{ri} \subset b_{r+1,j}$ in order, and so on recursively (Fig. 3). So it suffices to scan the features once (in the proper order) accumulating their mass to $L_1^k, \dots, L_{R-1}^k$. Whenever the visit crosses a bin boundary at some level r, the algorithm adds $\min\{L_r^1, L_r^2\}$ to F_r, resets L_r^1 and L_r^2 and moves on.[5]

3. Two novel RMKs

To illustrate the flexibility of the RMK construction, we introduce two new matching kernels.

Graph Matching Kernel. Graphs have been used extensively for representing and matching images. Usually a graph is constructed by connecting interest points or other features in structures such as constellations, and sketches (see for instance [4, 6, 12, 5] and references therein). Matching graphs however is difficult due to the high instability of such structures and the combinatorial complexity of the search. Roughly speaking, three approaches are used: (i) focus on simple structures (such as small graphs, trees or stars) that enable exhaustive search [6, 5], (ii) use statistical searching procedures (e.g. RANSAC, Swendsen-Wang

[3] This also suggests why counting the same features at multiple relaxation levels do not really introduce bias in the comparison

[4] The complexity is $O(NR)$ where $N = N_1 + N_2$ is the number of features from the two images to be compared and R is the number of relaxations. The algorithm is also space efficient as it requires only $O(N + R)$ memory.

[5] A further speed-up is obtained if features are pre-merged at the finer relaxation level B_0 before running the algorithm. This is especially useful for kernel such as PDK which compare pairs of interest point and may have large feature sets.

sampling [12]), and (iii) use approximated matching methods (e.g. spectral methods [18]).

Here we experiment with a loose but robust voting scheme, reminiscent of [10] and PDK, based on comparing interest point pairs. Consider a graph G whose nodes are interest points $l_1, \ldots, l_N$ with associated descriptors $d_1, \ldots, d_N$. Let $G = \{e_m, m = 1, \ldots, M\}$ be the collection of edges forming the graph, where $e_m = \{l_i, l_j\}$ is an (unordered) pair of image locations. Let ρ_{ij} be the graph distance from l_i to l_j (i.e. the length of the shortest path connecting l_i to l_j). We construct an RMK by considering triplets (d_i, d_j, ρ_{ij}) as the base features. We quantize the descriptor space as in PDK or SPMK (ρ has already a discrete structure) to obtain the base dictionary B_0. We then define the successive relaxation levels $B_1, \ldots, B_{R-1}$ by merging values of the index ρ, using the linear scheme of PDK/RMK. So the kernel has the form

$$K(G^1, G^2) = \sum_{r=0}^{R-1} w_r \sum_{(d_i, d_j, \rho) \in B_r} k(h^1_{B_r}(d_i, d_j, \rho), h^2_{B_r}(d_i, d_j, \rho)). \tag{4}$$

In the following we refer to this kernel as *Graph Matching Kernel* (GMK). GMK checks for the presence of edges between images features, as specified by the graph structure. Despite this fact, in the limit when all nodes have unique identifiers, $K(G^1, G^2)$ assumes its maximum value $\sum_{r=0}^{R-1} w_r$ if, and only if, $G^1 \equiv G^2$.

Agglomerative Information Bottleneck Kernel. As a second example of RMK, we introduce a kernel similar in spirit to PMK. We start by a basic quantization B_0 of the feature descriptors d_i (we discard the locations l_i). Then we define the successive relaxations B_r by iteratively merging bins of the base dictionary B_0. However, instead of guiding the merges based on descriptor similarity (as PMK does), we use Agglomerative Information Bottleneck (AIB, [20]) to obtain a sequence of binary merges. AIB produces a sequence of relaxations B_r so that the information $I(d, c)$ between the feature descriptor $d \in B_r$ (regarded as a random variable) and the class label c is maximally preserved. We use $w_r = I_r$ to penalize coarser relaxations which correspond to uninformative dictionaries, where I_r is the residual information $I(d, c)$ at the relaxation level r. We call this *Agglomerative Information Bottleneck Matching Kernel* (AIBMK).

4. Experiments

4.1. GMKs to match unstable graphs

The first experiment (Fig. 4) illustrates graph matching by GMK. Given an image I^k, we construct a graph as follows: we run Canny's edge detector on the image, we extract straight edge segments, and we complete the graph by constrained Delaunay triangulation. We then extract SIFT keys at the node locations (fixed window size and orientation) using software from [22] and we create a dictionary of only sixteen visual words (such a vocabulary is not very distinctive but quite invariant). This yields graphs G^1 and G^2 from the two images. We then select a location l_i in the first image and extract a subgraph $S^1(l_i) \subset G^1$, defined as the union of l_i with its neighbors at (graph) distance not greater than $T = 2$. Then we try to match $S^1(l_i)$ to $S^2(l_j)$ for all similarly constructed subgraphs in the second image. Notice the large variation in the structure of the graphs being matched, due both to instability of the construction of the image graphs G^k and the selection of the subgraphs S^k. We evaluate quantitatively how many subgraphs can be successfully matched in a test sequence from [15]. This data is devised to evaluate affine invariant descriptors; here we show that RMK is robust enough to match unstable interest points graphs.

4.2. RMKs for object recognition

We evaluate GMK, AIBMK, PDK, PDK/RMK in object recognition experiments on the Graz-02 and Pascal-05 datasets (mainly for the sake of comparison with previous related approaches). We also compare the methods against the baseline BoF as described by [25], which we summarize next. Each image is normalized so that the largest side measures 640 pixels. Then the Harris and Laplace operators are used to extract multiscale interest points using publicly available code from [3]. We remove features of scale below 2.5 pixels (we also remove duplicate features due to a bug in the software). As in [25], we fix the orientation of the patches to a nominal value (i.e. the interest points are not rotationally invariant). At each interest point we compute a SIFT descriptor. The visual vocabulary is formed by running k-means with $k = 200$ (Lloyd algorithm) for each category independently, and then joining the dictionaries. Bag of features are compared by the χ^2 RBF kernel, which performs better on average. The GMK, AIBMK, PDK, PDK/RMK also use the same χ^2 basis kernel and the RBF transformation. We use an SVM in all experiments. The parameter C of the SVM [19] is learned by 10-fold cross validation. The graph used in GMK is computed by Delaunay triangulation of the points (we do not extract edges).

For Graz-02 we use the same training and testing sets of [13, 21]. For Pascal-05 we use the training and validation sets from the challenge as training data and the test-2 (difficult) test set as testing data. Results are compared in Table 6 against [13, 21] and the winner of Pascal-05 VOC competition. ROC curves are reported in Fig. 5.

Our kernels are competitive, outperforming previous state of the art in four of the seven categories. Our implementation of PDK outperforms the original paper [13] in all but one cases, perhaps due to the fact that we use the χ^2 and RBF compbination. We also compare favorably to [21] and

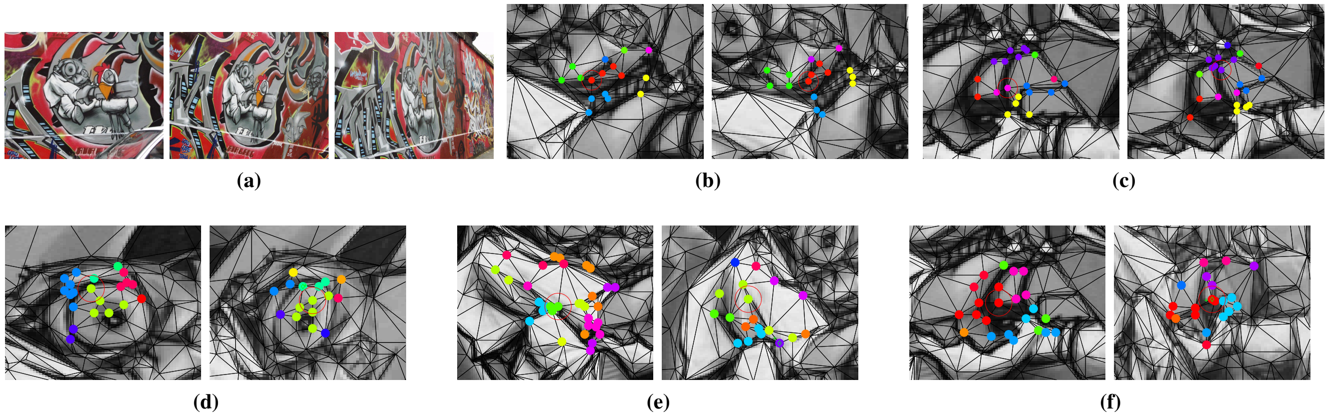
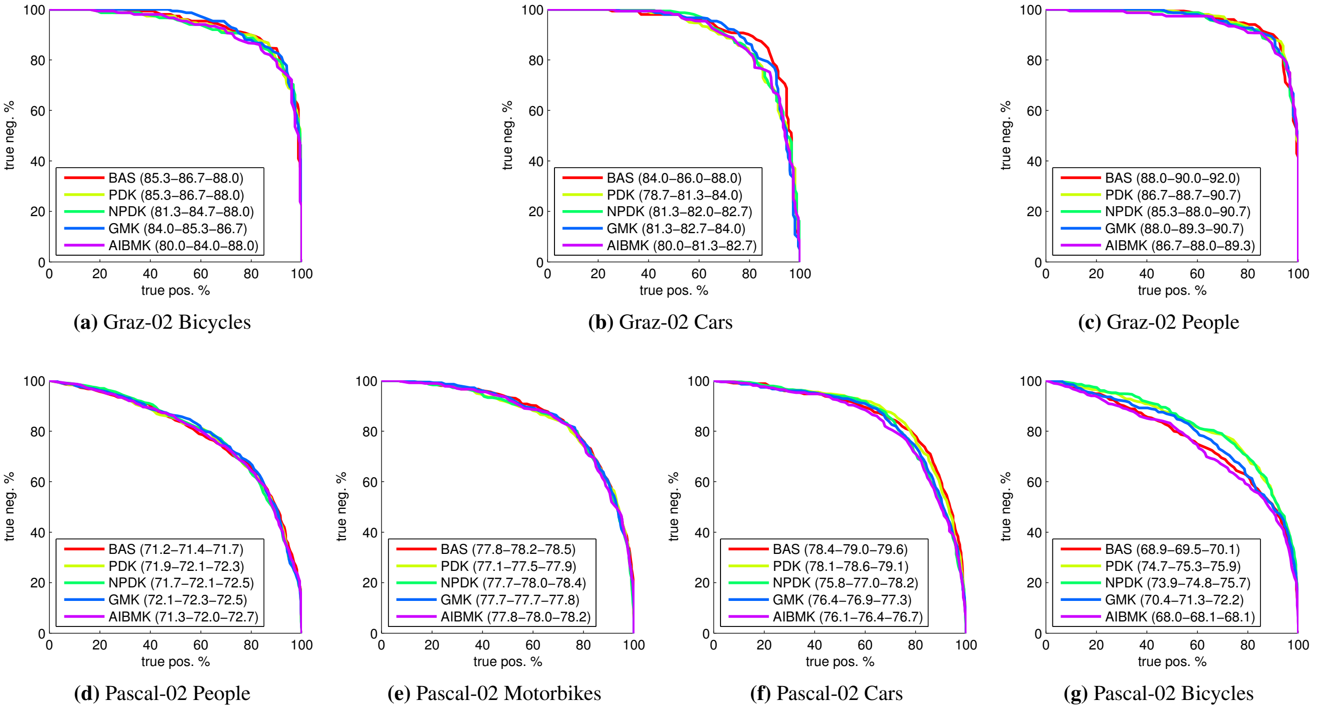


Figure 4: **GMK: robustness evaluation.** (a) A few images from [15]. The data consists of six image: a frontal view five other views from, 20 to 60 degrees of slant. Here we construct a graph by downsampling the images by half, computing a Canny edge map and running constrained Delaunay triangulation. We then compute SIFT features at nodes (fixed orientation and window size of 20 pixels). This construction is *not* affine invariant and the resulting graph is highly unstable. We make the node labels as invariant as possible by choosing a small dictionary size (64 bins). We then match each subgraph $S^1(l_i)$ in the frontal view to similar graphs $S^k(l_j)$ in the other views (we do not try to remove ambiguous matches). Using the ground truth homography, we record the graph distance from the center of the best matching subgraph to the actual reprojection. (b) a match at graph distance 0 from the 20^o views pair. (c) A match with graph distance 1 – the overlap is still very good. (d)-(f) two matches at 30^o. (f) A match at 50^o. Up to 20^o of slant 83% of the match are within graph distance 2. At 30^o this number reduces to 57%. After that the deformation of the descriptors is excessive and matching becomes unreliable.



(a) Graz-02 Bicycles

(b) Graz-02 Cars

(c) Graz-02 People

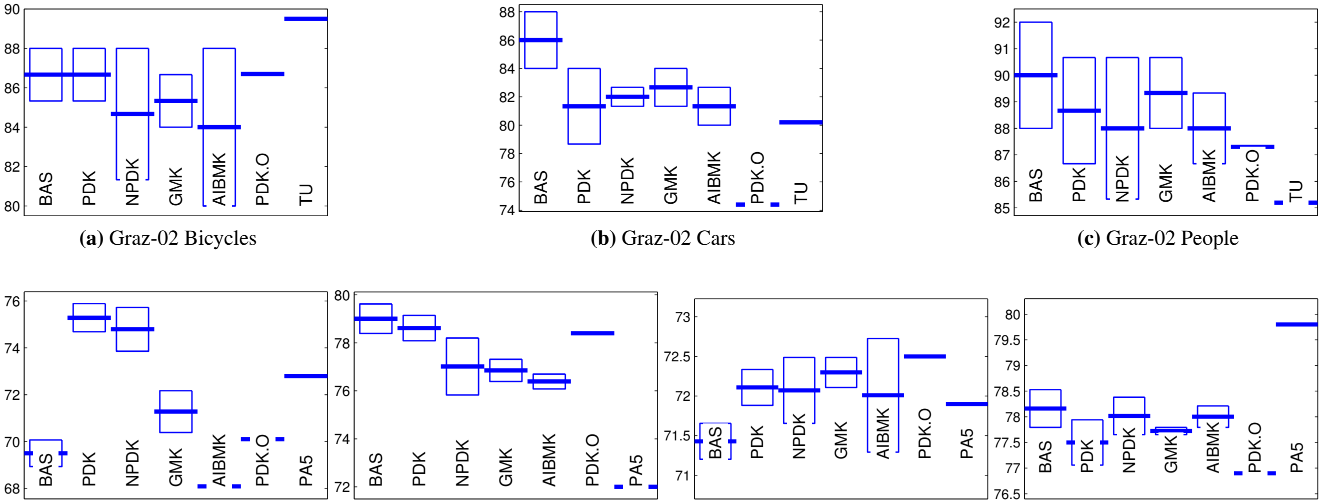
(d) Pascal-02 People

(e) Pascal-02 Motorbikes

(f) Pascal-02 Cars

(g) Pascal-02 Bicycles

Figure 5: **ROC curves for Pascal-05 and Graz-02.** We compare the average ROCs obtained in several runs of the various algorithms (we average ROC curves along lines from the origin; in this way the curve passes by the average equal-error-rate point).



(a) Graz-02 Bicycles

(b) Graz-02 Cars

(c) Graz-02 People

(d) Pascal-05 Bicycles

(e) Pascal-05 Cars

(f) Pascal-05 People

(g) Pascal-05 Motorbikes

Figure 6: **Equal Error Rates for Pascal-05 and Graz-02.** We report the maximum, minimum and average EER for each algorithm in multiple runs (as the construction of the dictionary is randomized). The variability, especially in the smaller Graz-02 dataset, is relatively large. This makes it difficult to compare directly to previous work, which do not report this information. Here PDK.O refers to [13], TU to [21] and PA5 to Pascal-05 winner. All algorithms, whether they use spatial information or not, are very close. The baseline algorithm performs as well or better in many of the cases, and it is very close to the best algorithm in the others. Makes exception Pascal-05 bikes, where we were able to obtain the better performance by method exploiting the spatial structure.

the Pascal-05 winner.

We should note, however, than in most cases the advantage of one method on another is small (see for instance GR-bicycles). In particular, the baseline algorithm performs in practice as well and in some case better than these more sophisticated kernels and [21] (which uses dense features and a large vocabulary as opposed to sparse feature and a small vocabulary).

5. Conclusions

We have introduced RMK as a generalization of popular kernels for image categorizations. The formulation defines a large space of possible useful kernels, and suggests modifications and improvements to the current ones. We also have introduced a novel interpretation of the kernel weights and showed the monotonicity property of the relaxed similarity scores (1). These observations transfer directly to previous method as well.

We have introduced two new examples of RMKs: the GMK and AIBMK kernels. GMK have been demonstrated successfully for matching graphs of features in a wide-baseline matching experiment. We also have tested our kernels on object categorization on Pascal-05 and Graz-02. However, we also noticed that a baseline BoF formulation is often as competitive, which, we hope, will stimulate a useful debate in the community.

Acknowledgments. Research supported by ONR 67F-1080868 and AFOSR FA9550-06-1-0138.

A. Appendix

We study the parametric family of kernels among histograms given by $K(p,q) = \sum_i k_{\alpha|\beta}(p_i,q_i)$, where [9]

$$k_{\alpha|\beta} = \frac{p_i+q_i}{2} - \frac{1}{2Z}\left[\left(\frac{p_i^\alpha+q_i^\alpha}{2}\right)^{\frac{1}{\alpha}} - \left(\frac{p_i^\beta+q_i^\beta}{2}\right)^{\frac{1}{\beta}}\right] \tag{5}$$

where $\alpha \geq 1$ and $\beta \in [-\infty,-1]\cup[\frac{1}{2},\alpha]$ and the normalization constant Z is equal to $2^{-\frac{1}{\alpha}} - 2^{-\frac{1}{\beta}}$ if $\beta > 0$ and to $2^{-\frac{1}{\alpha}}$ if $\beta < 0$. l_1, Hellinger's and χ^2 kernels are obtained for (α,β) equal to $(\infty,1)$, $(1,\frac{1}{2})$ and $(1,-1)$ respectively. In the following we restrict to the case $\beta \leq 1$ (we verified by simulation that these results do not always hold if $\beta > 1$).

Lemma 1. *Let $x_1,x_2,y_1,y_2 \in \mathbb{R}_+$ be non negative numbers and let $k = k_{\alpha|\beta}$ as defined above. Moreover, let $\beta \leq 1$. Then*

$$k(x_1+x_2,y_1+y_2) \geq k(x_1,y_1) + k(x_2,y_2)$$

Proof. Let $f_\alpha(x_i,y_i) = (x_i^\alpha + y_i^\alpha)^{1/\alpha}$. Since $\alpha \geq$

1, by Minkowsky's inequality[6] $f_\alpha(x_1 + x_2, y_1 + y_2) \leq f_\alpha(x_1, y_1) + f_\alpha(x_2, y_2)$. Minkowsky's inequality reverses when the exponent is smaller than 1, for which $f_\beta(x_1 + x_2, y_1 + y_2) \leq f_\beta(x_1, y_1) + f_\beta(x_2, y_2)$. Substituting back in (5) we obtain the desired inequality. □

Theorem 1 (Monotonicity of the kernel)**.** *Let $p, q \in \mathbb{R}^n_+$ be non-negative real vectors. Let W be a stochastic matrix (i.e. $W \in \mathbb{R}^{m\times n}_+$, $\mathbf{1}^\top W = \mathbf{1}^\top$). Let $K(p, q)$ defined as above, with $\beta \leq 1$.Then*

$$K(Wp, Wq) \geq K(p, q).$$

Proof. We have

$$K(Wp, Wq) = \sum_i K\left(\sum_j w_{ij} p_j, \sum_j w_{ij} q_j\right)$$

Applying iteratively the lemma $n - 1$ times yields

$$K(Wp, Wq) \geq \sum_i \sum_j K(w_{ij} p_j, w_{ij} q_j).$$

But K is homogeneous (i.e. $K(cx, cy) = cK(x, y)$), so

$$K(Wp, Wq) \geq \sum_j \sum_i w_{ij} K(p_j, q_j) = K(p, q).$$

□

References

[1] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *PAMI*, 11(6):567–585, 1989.

[2] G. Csurka, C. R. Dance, L. Dan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc. ECCV*, 2004.

[3] G. Dorkó. Scale and affine invariant local detectors and descriptors. Technical report, INRIA, 2005.

[4] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proc. ICCV*, 2003.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005.

[6] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proc. CVPR*, 2005.

[7] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative calssification with sets of image features. Technical Report MIT-CSAIL-TR-2006-020, MIT, 2006.

[8] C. Guo, S.-C. Zhu, and Y. N. Wu. Towards a mathematical theory of primal sketch and sketchability. In *Proc. ICCV*, page 1228, 2003.

[9] M. Hein and O. Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In *Proc. AISTAT*, 2005.

[10] H. Kashima and A. Inokuchi. Kernels for graph classification. In *ICDM Workshop on Active Mining*, 2002.

[11] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, 2006.

[12] L. Lin, S.-C. Zhu, and Y. Wang. Layered graph matching with graph editing. In *Proc. CVPR*, 2007.

[13] H. Ling and S. Soatto. Proximity distribution kernels for geometric context in category recognition. In *Proc. CVPR*, 2007.

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2(60):91–110, 2004.

[15] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 1(60):63–86, 2004.

[16] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *Proc. ECCV*, 2006.

[17] F. Perronnin and C. Dance. Fisher kenrels on visual vocabularies for image categorizaton. In *Proc. CVPR*, 2006.

[18] H. Qiu and E. R. Hancock. Graph matching and clustering using spectral partitions. *Pattern Recognition*, 39, 2006.

[19] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.

[20] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *Proc. NIPS*, 1999.

[21] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *Proc. ICCV*, 2007.

[22] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://vision.ucla.edu/vlfeat`, 2008.

[23] A. Vedaldi and S. Soatto. Features for recognition: Viewpoint invariance for non-planar scenes. In *Proc. ICCV*, 2005.

[24] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. ECCV*, 2000.

[25] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 2006.

[6] I.e. by th l^α-triangle inequality applied to vectors (x_1, y_1) and (x_2, y_2).