10-1-2004

Mercer Kernels for Object Recognition with Local Features

Siwei Lyu
Dartmouth College

Mercer Kernels for Object Recognition with Local Features

Siwei Lyu
Department of Computer Science
Dartmouth College
Hanover NH 03755

A new class of kernels for object recognition based on local image feature representations are introduced in this paper. Formal proofs are given to show that these kernels satisfy the Mercer condition. In addition, multiple types of local features and semilocal constraints are incorporated. Experimental results of SVM classifiers coupled with the proposed kernels are reported on recognition tasks with the COIL-100 database and compared with existing methods. The proposed kernels achieved competitive performance and were robust to changes in object configurations and image degradations.

Correspondence should be addressed to S. Lyu. 6211 Sudikoff Lab, Department of Computer Science, Dartmouth College, Hanover NH 03755. tel: 603.643.8722; email: lsw@cs.dartmouth.edu.

1. Introduction

Kernel methods received attention originally as a "trick" to introduce non-linearity into support vector machines (SVM) [21]. Evaluating a kernel function between two data is equivalent to computing the scalar product of their images in a non-linearly mapped space (usually termed as feature space). It is realized later that kernel methods are more general. Similar to SVM, many linear algorithms (e.g., PCA and Fisher linear discriminant) depend on data through their scalar products. By substituting the scalar products with kernel evaluations, these algorithms can discover non-linear patterns in data. At the same time, they are still computationally efficient, as the kernel function is evaluated in the input space [20]. Instead of using general-purpose kernels (e.g., Gaussians), recent effort has been put on designing kernels tailored to the requirements of a specific application. Such kernels better reflect the similarities between data and thus incorporate more domain knowledge into the algorithm.

One important application of kernel method is appearance based object recognition. Object recognition remains one of the most challenging problems in computer vision. Changes in illumination, pose, viewing angle, occlusion, clutters and non-rigid deformations are just a few of the complicated problems a recognition system has to face. Many applications of kernel methods to object recognition are based on global image features (e.g., global grayvalue histograms) [14, 4, 15]. Though promising performance has been reported, these methods are plagued by the deficiencies of the global features, such as being sensitive to image degradations (e.g., noise, occlusion and background clutters) and not robust under changes in object configurations (e.g., translation and scaling).

Recent years have seen impressive developments in using local features computed at interest points for matching and recognition [9, 17, 16, 10, 2]. Such approaches lead to robust and compact image representations that lend themselves to powerful pattern analysis algorithms. However, the local feature representations pose several challenges to kernel design. First, it requires the kernel to work efficiently on inputs of variable lengths, as images may have a different number of local features. Secondly, the kernel should measure similarity of two unordered sets of local features, where no explicit correspondence is available. Furthermore, several different types of local features are usually collected and they need to be fused into the kernel. For better performance, semilocal spatial and geometrical constraints between interest points should also be incorporated. Finally, to guarantee unique global optimal solutions for the SVM algorithm, the kernel must also satisfy the Mercer condition. Unfortunately, existing methods (e.g., [1, 22, 23, 12, 8]) are not satisfactory in that they do not meet all of these requirements.

The major contribution of this paper is the definition of a new class of kernels for object recognition, based on local feature representations. Formal proofs are given to show that this class of kernels satisfy the Mercer condition and reflect similarities between sets of local features. In addition, multiple local feature types and semilocal constraints are incorporated to reduce mismatches between local features, thus further improve the classification performance. Results are shown on testing the proposed kernels, coupled with SVM classification, on recognition tasks with the COIL-100 database.

2. Methods

In this section, after a brief review of Mercer kernels and local features, the proposed kernel is described and compared with previous approaches. Then kernels using multiple types of local features and semilocal constraints are introduced, followed by an algorithm summarizing the overall process.

2.1. Mercer kernel

Admissible kernel functions satisfy the Mercer condition (hence usually termed as Mercer kernels). For an input space $\mathcal{X}$, if there is a mapping $\phi : \mathcal{X} \rightarrow \mathcal{H}$ that maps any $x, z \in \mathcal{X}$ into a Hilbert space $\mathcal{H}$, then a Mercer kernel, $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$, is constructed as $K(x, z) = \langle \phi(x), \phi(z) \rangle_{\mathcal{H}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the scalar product operator in $\mathcal{H}$. Such a function K satisfy the Mercer condition, an equivalent description of which is stated formally in the following proposition:

Proposition 1 (Theorem 3.11, [20]) *Let $\mathcal{X}$ be any input space and $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$ a symmetric function, K is a Mercer kernel if and only if the kernel matrix formed by restricting K to any finite subset of $\mathcal{X}$ is positive semi-definite (having no negative eigenvalues).*

The Mercer condition is essential to kernel design, as it is the key requirement for a unique global optimal solution to the kernel-extended pattern analysis algorithms based on convex optimization (e.g., SVM) [20].

Instead of using its definition, a Mercer kernel is usually constructed in other more convenient ways. For data in a vector space, one can choose from standard off-the-shelf kernels, a common choice being a Gaussian:

$$K_R(x, z) \quad = \quad \exp(-\tfrac{\|x-z\|^2}{2\sigma^2}). \qquad (1)$$

There are also Mercer kernels designed specifically for structured data types, such as strings, trees or graphs [20]. In addition, new Mercer kernels can be built on existing ones. Several properties of Mercer kernels relevant to this aspect are summarized in the following proposition:

Proposition 2 *For Mercer kernels, the following facts hold*

(i) **(Proposition 2.22, [20])** *The product of two Mercer kernels is a Mercer kernel. Thus, a monomial of any degree of a Mercer kernel is a Mercer kernel.*

(ii) **(Lemma 1, [7])** *Let K be a Mercer kernel defined on $\mathcal{X} \times \mathcal{X}$, for any finite $A, B \subseteq \mathcal{X}$, define $\widetilde{K}(A,B) = \sum_{x\in A}\sum_{y\in B} K(x,y)$. Then $\widetilde{K}$ is a Mercer kernel on $2^{\mathcal{X}} \times 2^{\mathcal{X}} \setminus \{\emptyset\}$.*

(iii) **(Proposition 11.75, [20])** *For a data space $\mathcal{X}$ that can be decomposed as $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_N$ and $x = (x_1, \cdots, x_N) \in \mathcal{X}$, for which $x_i \in \mathcal{X}_i$, denote Mercer kernels K_i on $\mathcal{X}_i \times \mathcal{X}_i$, $i = 1, \cdots, N$. For $x, z \in \mathcal{X}$, $K(x,z) = \prod_{i=1}^{N} K_i(x_i, z_i)$, is a Mercer kernel on $\mathcal{X} \times \mathcal{X}$. Such a kernel is a special case of the $\mathcal{R}$-convolution kernel [7].*

Besides satisfying the Mercer condition, many applications also require the designed kernel to reflect similarities between the data being studied. As kernels are elicited from scalar products, they are expected to have larger values for data that are more similar to each other.[1] Admissible kernel functions satisfy the Mercer condition (hence usually termed as Mercer kernels). For an input space $\mathcal{X}$, if there is a mapping $\phi : \mathcal{X} \to \mathcal{H}$ that maps any $x, z \in \mathcal{X}$ into a Hilbert space $\mathcal{H}$, then a Mercer kernel, $K : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$, is constructed as $K(x,z) = \langle \phi(x), \phi(z) \rangle_{\mathcal{H}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the scalar product operator in $\mathcal{H}$. Such a function K satisfy the Mercer condition, an equivalent description of which is stated formally in the following proposition:

Proposition 1 (Theorem 3.11, [20]) *Let $\mathcal{X}$ be any input space and $K : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$ a symmetric function, K is a Mercer kernel if and only if the kernel matrix formed by restricting K to any finite subset of $\mathcal{X}$ is positive semi-definite (having no negative eigenvalues).*

The Mercer condition is essential to kernel design, as it is the key requirement for a unique global optimal solution to the kernel-extended pattern analysis algorithms based on convex optimization (e.g., SVM) [20].

Instead of using its definition, a Mercer kernel is usually constructed in other more convenient ways. For data in a vector space, one can choose from standard off-the-shelf kernels, a common choice being a Gaussian:

$$K_R(x,z) \quad = \quad \exp(-\tfrac{\|x-z\|^2}{2\sigma^2}). \qquad (2)$$

There are also Mercer kernels designed specifically for structured data types, such as strings, trees or graphs [20]. In addition, new Mercer kernels can be built on existing ones. Several properties of Mercer kernels relevant to this aspect are summarized in the following proposition:

Proposition 2 *For Mercer kernels, the following facts hold*

(i) **(Proposition 2.22, [20])** *The product of two Mercer kernels is a Mercer kernel. Thus, a monomial of any degree of a Mercer kernel is a Mercer kernel.*

(ii) **(Lemma 1, [7])** *Let K be a Mercer kernel defined on $\mathcal{X} \times \mathcal{X}$, for any finite $A, B \subseteq \mathcal{X}$, define $\widetilde{K}(A,B) = \sum_{x\in A}\sum_{y\in B} K(x,y)$. Then $\widetilde{K}$ is a Mercer kernel on $2^{\mathcal{X}} \times 2^{\mathcal{X}} \setminus \{\emptyset\}$.*

(iii) **(Proposition 11.75, [20])** *For a data space $\mathcal{X}$ that can be decomposed as $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_N$ and $x = (x_1, \cdots, x_N) \in \mathcal{X}$, for which $x_i \in \mathcal{X}_i$, denote Mercer kernels K_i on $\mathcal{X}_i \times \mathcal{X}_i$, $i = 1, \cdots, N$. For $x, z \in \mathcal{X}$, $K(x,z) = \prod_{i=1}^{N} K_i(x_i, z_i)$, is a Mercer kernel on $\mathcal{X} \times \mathcal{X}$. Such a kernel is a special case of the $\mathcal{R}$-convolution kernel [7].*

Besides satisfying the Mercer condition, many applications also require the designed kernel to reflect similarities between the data being studied. As kernels are elicited from scalar products, they are expected to have larger values for data that are more similar to each other.[2]

2.2. Local feature representation

Local features are localized descriptors that provide distinct information about a specific location of an image. Many local features (e.g., [9, 17, 16, 10, 2]) are designed to be invariant under certain image transformations, such as rotation and scaling, so that they are relatively stable to changes in object configurations. Local features have proved to be very successful in appearance based object matching and recognition, as they are distinctive, robust to image degradation and transformation, and require no segmentation [11].

Local features are usually collected at or in the neighboring region around interest points, which are specific positions in an image that carry distinctive features of the object being studied. Interest points are found by an interest point detector, popular choices for which are the Harris detector [6] and multi-resolution based detectors [19].

In this paper, we denote $p_i = (x_i, y_i)$ as the coordinate (in the image plane) of the i-th interest point detected in the image, and vector F_i as the local feature computed at or around p_i. An image I_a is represented by the set of local features corresponding to all interest points detected, denoted as $\mathcal{F}_a = \{F_1^{(a)}, \cdots, F_{|\mathcal{F}_a|}^{(a)}\}$.

2.3. Related work

With local feature representation, an image is concisely represented by its set of local feature vectors. Accordingly, kernels that match images could be defined between two sets of local feature vectors. We start by enumerating some desirable properties of such kernels:

[1] Strictly speaking, normalized kernel evaluates the cosine similarity of two mapped data in the feature space.

[2] Strictly speaking, normalized kernel evaluates the cosine similarity of two mapped data in the feature space.

- The kernel should satisfy the Mercer condition;
- The computation of the kernel should be efficient in both time and space;
- The kernel should be able to handle inputs with variable lengths, as the number of interest points may vary across different images;
- The kernel should reflect similarities between two sets of local feature vectors.

It should be noted that the local feature representation does not provide correspondence between local features of two images, while only the correctly matched local features carry meaningful discriminant information. However, finding the optimal matching of local features is not always feasible in practice and many algorithms are based on heuristics. One important assumption common to most matching algorithms is that the correctly matched local features are more similar to each other than otherwise.

These properties void the use of off-the-shelf kernels, such as a Gaussian, as the underlying data (sets of vectors) are not from a vector space. One can normalize the length of inputs by padding zeros. Whereas the inputs can be assumed in a vector space now, the computed quantity is of little interest to recognition. Notice, however, that it is relatively easy to build a kernel K_F on the local features, as they are vectors with identical dimensions. A natural idea is to construct composite kernels on the basis of such kernels, which work with sets of local features.

One simple example of such an approach is the summation kernel. On two local feature sets, $\mathcal{F}_a = \{F_1^{(a)}, \cdots, F_{|\mathcal{F}_a|}^{(a)}\}$ and $\mathcal{F}_b = \{F_1^{(b)}, \cdots, F_{|\mathcal{F}_b|}^{(b)}\}$, of two images, I_a and I_b, the summation kernel is defined as

$$K_S(\mathcal{F}_a, \mathcal{F}_b) = \frac{1}{|\mathcal{F}_a|}\frac{1}{|\mathcal{F}_b|}\sum_{i=1}^{|\mathcal{F}_a|}\sum_{j=1}^{|\mathcal{F}_b|} K_F(F_i^{(a)}, F_j^{(b)}). \quad (3)$$

Simple application of Proposition 2, part (ii), shows that the summation kernel satisfies the Mercer condition. However, its discriminative ability is compromised by the fact that all possible matchings between local features are combined with equal bias. The good matchings, highly out-numbered, could be easily swamped by the bad ones.

In [22], a kernel function based on matching local features was proposed

$$K_M(\mathcal{F}_a, \mathcal{F}_b) = \frac{1}{2}\frac{1}{|\mathcal{F}_a|}\sum_{i=1}^{|\mathcal{F}_a|}\max_{j=1,\cdots,|\mathcal{F}_b|} K_F(F_i^{(a)}, F_j^{(b)}) + \frac{1}{2}\frac{1}{|\mathcal{F}_b|}\sum_{j=1}^{|\mathcal{F}_b|}\max_{i=1,\cdots,|\mathcal{F}_a|} K_F(F_j^{(b)}, F_i^{(a)}). \quad (4)$$

Function K_M has the desired property of reflecting similarities of two sets of local feature vectors, as it only considers the similarities of the best matched local features. Unfortunately, despite the claim in [22], K_M is not a Mercer kernel, for which a detailed proof is given in Appendix A. In [1], a similar non-Mercer kernel based on a sub-optimal matching between local features is used but measures are provided so that the probability of the kernel not being positive semi-definite is bounded. However, as pointed out earlier, the Mercer condition is essential to reliable recognition, Mercer kernels are still preferable in practice.

In [23], a Mercer kernel is proposed for sets of vectors based on the concept of principal angles between two linear subspaces. However, this kernel showed poor recognition performance as reported in [5]. In [8], the Bhattacharyya kernel is introduced where a set of vectors is represented as a multivariate Gaussian. Though provably satisfying the Mercer condition, evaluating this kernel is cubic in the number of local features. Furthermore, good matchings do not necessarily distinguish themselves in such a setting. In [12], a kernel based on Kullback-Leibler divergence is proposed. However, as the authors pointed out, it is not clear if such a kernel satisfies the Mercer condition.

2.4. A Mercer kernel between local feature sets

As discussed earlier, only the correctly matched local features with large similarity measures provide meaningful discriminant information for recognition. This indicates that such matched pairs should dominate in the kernel evaluation, if we expect the kernel to measure similarities between two sets of local feature vectors. However, directly summing the maximum similarities as in the case of K_M results in inadmissible kernels that violate the Mercer condition.

In this paper, a new class of kernels are proposed that measure similarity between local feature sets and that provably satisfy the Mercer condition. The proposed kernel function is defined as

$$K_{\mathcal{F}}(\mathcal{F}_a, \mathcal{F}_b) = \frac{1}{|\mathcal{F}_a|}\frac{1}{|\mathcal{F}_b|}\sum_{i=1}^{|\mathcal{F}_a|}\sum_{j=1}^{|\mathcal{F}_b|}\left[K_F(F_i^{(a)}, F_j^{(b)})\right]_{,}^{p} \quad (5)$$

where integer $p \geq 1$ is the kernel parameter. With $p = 1$, the proposed kernel includes the summation kernel as a special case. Similar to the summation kernel, all possible matchings between local features in the two sets are considered in $K_{\mathcal{F}}$, but with different bias. It is through the kernel parameter p that the correct matched local features are given dominant bias in $K_{\mathcal{F}}$. This is made more clear if $K_{\mathcal{F}}$ is rewritten as

$$K_{\mathcal{F}}(\mathcal{F}_a, \mathcal{F}_b) = \frac{1}{2}\frac{1}{|\mathcal{F}_a|}\sum_{i=1}^{|\mathcal{F}_a|}\frac{1}{|\mathcal{F}_b|}\sum_{j=1}^{|\mathcal{F}_b|}\left[K_F(F_i^{(a)}, F_j^{(b)})\right]^p + \frac{1}{2}\frac{1}{|\mathcal{F}_b|}\sum_{j=1}^{|\mathcal{F}_b|}\frac{1}{|\mathcal{F}_a|}\sum_{i=1}^{|\mathcal{F}_a|}\left[K_F(F_j^{(b)}, F_i^{(a)})\right]_{.}^{p} \quad (6)$$

Now $K_{\mathcal{F}}$ has a similar form as K_M: both are sums of some similarity measures over each local feature vector. Only the max function in K_M is replaced here with a summation of monomials. Consider a local feature $F_i^{(a)}$ of $\mathcal{F}_a$ (though the following results are also true for members of $\mathcal{F}_b$), and its $|\mathcal{F}_b|$ kernel evaluations with each member of $\mathcal{F}_b$, $K_F(F_i^{(a)}, F_1^{(b)}), \cdots, K_F(F_i^{(a)}, F_{|\mathcal{F}_b|}^{(b)})$. The similarity between $F_i^{(a)}$ and local feature set $\mathcal{F}_b$ is measured in Equation (6) as

$$\frac{1}{|\mathcal{F}_b|}\sum_{j=1}^{|\mathcal{F}_b|}\left[K_F(F_i^{(a)}, F_j^{(b)})\right]^p. \tag{7}$$

Without loss of generality, let us assume $K_F(F_i^{(a)}, F_1^{(b)}) \geq \cdots \geq K_F(F_i^{(a)}, F_{|\mathcal{F}_b|}^{(b)})$. The contribution of the best matched local feature in $\mathcal{F}_b$ with $F_i^{(a)}$ in the sum of Equation (7) is:

$$\kappa = \left[K_F(F_i^{(a)}, F_1^{(b)})\right]^p / \sum_{j=1}^{|\mathcal{F}_b|}\left[K_F(F_i^{(a)}, F_j^{(b)})\right]^p. \tag{8}$$

The larger the value of p is, the more dominant is the best matched pair. As p approaches infinity, all but the maximal values will have a negligible fraction in the sum.

Furthermore, if we require that the similarity of the best matched pair in the sum has a fraction above a given threshold ρ, a lower bound of p can be computed as:

$$p \geq \log\frac{1-\rho}{(|\mathcal{F}_b|-1)\rho} / \log\frac{K_F\left(F_i^{(a)}, F_2^{(b)}\right)}{K_F\left(F_i^{(a)}, F_1^{(b)}\right)}, \tag{9}$$

where $F_2^{(b)}$ is the second best matching local feature in $\mathcal{F}_b$ for $F_i^{(a)}$ (a detailed proof is given in Appendix B). A proper p can be chosen as the maximum of such lower bounds over all training data.

The proposed kernel satisfies the Mercer condition, which is formally stated in the following proposition:

Proposition 3 *Function $K_{\mathcal{F}}$ defined in Equation (5) is a Mercer kernel, if function K_F is a Mercer kernel defined on the local feature vectors.*

Proof First, note that $K_{\mathcal{F}}$ is symmetric by definition. Also $[K_F(\cdot,\cdot)]^p$ is a monomial of a Mercer kernel. With Proposition 2, part (i), it is also a Mercer kernel. Finally, $K_{\mathcal{F}}$ is constructed in way of Proposition 2, part (ii), therefore, it satisfies the Mercer condition.

2.5. Multiple local feature types

So far, in constructing kernels on local feature sets, only one type of local feature is considered . However, it is usually possible to compute multiple types of local features at an interest point. As each individual type of local feature may carry distinctive information about the underlying object, it is desirable to have them fused into the designed kernel. Hereafter, we will refer to each type of local feature as a base local feature.

Assume L different base local features are employed, and denote $f_l^{(a)} \in \mathcal{R}^{d_l}$ as the d_l-dimensional vector of the l-th base local feature computed at an interest point p_a, for $l = 1, \cdots, L$. Also assume that the similarity of the l-th base local feature is properly measured by a Mercer kernel, $K_f^{(l)}$.[3] The local feature of p_a is a vector of dimension $\sum_{l=1}^{L} d_l$, formed by stacking all $f_l^{(a)}$s as $F_a = ({f_1^{(a)}}^T, \cdots, {f_L^{(a)}}^T)^T$. A kernel between two such local features, F_a and F_b, is define as

$$K_F(F_a, F_b) = \prod_{l=1}^{L} K_f^{(l)}(f_l^{(a)}, f_l^{(b)}). \tag{10}$$

The function K_F satisfies the Mercer condition, Proposition 2, part (iii). It can then be substituted into the definition of $K_{\mathcal{F}}$, Equation 5, which now incorporates multiple types of local features.

2.6. Semilocal constraints

One problem of representing an image as an unordered set of local feature vectors is that such a representation is independent to the spatial locations of the interest points. Different objects, therefore, with similar local feature vectors laid out differently in the image plane are indistinguishable. On the other hand, as supported by the experimental results in [17], there are strong spatial correlations between the interest points and their corresponding local features in an image. Such correlations are termed semilocal constraints in [17]. For better recognition performance, it may be desirable to enforce such semilocal constraints in kernel design.

Following the method in [17], we use the local shape configuration to enforce semilocal constraints.[4] Specifically, an image is represented as a set of semilocal groups, which bundle together image information around spatially close interest points. One semilocal group is formed around each interest point (its central interest point) detected in an image. Each semilocal group is defined as a two component tuple, denoted as $g = \{\mathcal{F}, \Theta\}$. The first component, $\mathcal{F} = \{F_0, F_1, \cdots, F_k\}$, is a set of local features collected at the central interest point as well as its k-nearest neighbors, $p'_1, \cdots, p'_k$. The second component, $\Theta = (\theta_1, \cdots, \theta_k)$, is a vector containing neighboring angles in the constellation spanned by the central interest point and its k-nearest neighbors, Figure 1. These neighboring angles convey the local geometrical constraints within the semilocal group. As pointed out in [17], if we suppose that the transformations of objects can be locally approximated by a similarity transformation, then these angles have to be locally consistent.

[3] Such kernels are termed as minor kernels in [5]. In [22], several minor kernels for some state-of-the-art local feature representations are listed.

[4] We are reluctant to use positions of interest points directly in the kernel, as in [22]. Such a setting makes the kernel vulnerable to changes in the spatial configurations of the object (e.g., translation).

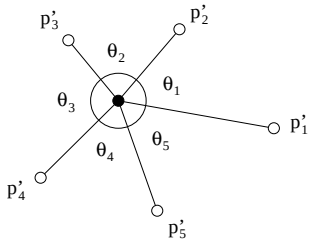

Figure 1: An example of semilocal group formed by an interest point (central filled dot) and its five nearest neighbors, $p'_1, \cdots, p'_5$. Hypothetical lines are added to show the neighboring angles.

An image I_a is now represented by a set of semilocal groups, $G_a = \{g_1^{(a)}, \cdots, g_{|G_a|}^{(a)}\}$. Correspondingly, the kernel matching images are now defined on two sets of semilocal groups. Similar to the approach taken in constructing kernel $K_{\mathcal{F}}$, we define a kernel between two sets of semilocal groups as

$$K_G(G_a, G_b) = \frac{1}{|G_a|}\frac{1}{|G_b|}\sum_{i=1}^{|G_a|}\sum_{j=1}^{|G_b|}\left[K_g(g_i^{(a)}, g_j^{(b)})\right]_{,}^{p} \quad (11)$$

where K_g is a Mercer kernel between two semilocal groups to be specified later, and integer p is the kernel parameter. A similar proof as that of Proposition 3 will show that kernel K_G satisfies the Mercer condition. Correct correspondence is still an important issue, as in the case of local features, only correctly matched semilocal groups are meaningful for recognition. The kernel parameter p in K_G has a similar role as its counterpart in kernel $K_{\mathcal{F}}$, which gives preference to good matchings between semilocal groups.

2.7. A circular-shift invariant kernel

In constructing K_G, Equation (11), kernel K_g between two semilocal groups is left unspecified. As a semilocal group consists of two parts, a natural way to design K_g is to use the product of two kernels individually defined on the two composing parts of g as:

$$K_g(g_a, g_b) = K_{\mathcal{F}}(\mathcal{F}_a, \mathcal{F}_b)K_{\mathcal{G}}(\Theta_a, \Theta_b), \quad (12)$$

where $K_{\mathcal{F}}$ is defined as in Equation (5). Kernel $K_{\mathcal{G}}$ is defined between two vectors of neighboring angles in the semilocal constellation. Special care is required to design such a kernel, as Θ is invariant under circular-shifts. For instance, consider again the example shown in Figure 1. A vector of neighboring angles as $(\theta_3, \theta_4, \theta_5, \theta_1, \theta_2)$ represents the same geometrical configuration as $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$. For this reason, kernel $K_{\mathcal{G}}$, which measures the similarity between two vectors of neighboring angles, should not treat such two vectors as different (i.e., it should also be invariant under circular-shifts).

In the most general setting, for two n-dimensional vectors $x = (x_0, \cdots, x_{n-1})^T \in \mathcal{R}^n$ and $y = (y_0, \cdots, y_{n-1})^T \in \mathcal{R}^n$, formally we define function $c : \mathcal{R}^n \times \{0, \cdots, n-1\} \to \mathcal{R}^n$ to be the circular-shift operator as $(c(y,l))_i = (y)_{(i+l) \bmod n}$, where $(y)_i$ is the i-th component of y and $0 \le l, i \le n-1$. Now consider function

$$K_{\mathcal{G}}(x,y) = \sum_{l=0}^{n-1}\left[K(x, c(y,l))\right]_{,}^{p} \quad (13)$$

where $K : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}$ is a Mercer kernel and satisfies that $K(x,y) = K(c(x,d), c(y,d))$ for $0 \le d \le n-1$. Many commonly used kernel functions (e.g., Gaussian) are valid candidates for K and in our case, we simply choose it to be the vector scalar product in $\mathcal{R}^n$ as $K(x,y) = x^T y$.

Proposition 4 *Function $K_{\mathcal{G}}$ as defined in Equation (13) has the following properties:*

(i) it is a Mercer kernel on $\mathcal{R}^n \times \mathcal{R}^n$;

(ii) it is invariant under circular-shifts, as for $x, y \in \mathcal{R}^n$, $K_{\mathcal{G}}(x,y) = K_{\mathcal{G}}(c(x,d_1), c(y,d_2))$, for $0 \le d_1, d_2 \le n-1$.

A full proof of these results is given in Appendix C.

Notice that in constructing kernel $K_{\mathcal{G}}$, we again employ a kernel parameter p to give dominant bias to good matchings. Finally, as both $K_{\mathcal{F}}$ and $K_{\mathcal{G}}$ satisfy the Mercer condition, according to Proposition 2, part (iii), their product K_g, Equation (12), is also a Mercer kernel.

2.8. Summary

The process of constructing a kernel for object recognition, as proposed in this paper, built with multiple types of local features and semilocal constraints, is summarized in the following algorithm:

1. With minor kernels K_f defined on base local features, construct kernel K_F on local features with Equation (10);
2. Construct kernel $K_{\mathcal{F}}$ on local feature sets with Equation (5);
3. Obtain a vector of neighboring angles in a semilocal group, and construct kernel $K_{\mathcal{G}}$ with Equation (13);
4. Combine kernels $K_{\mathcal{F}}$ and $K_{\mathcal{G}}$ into kernel K_g with Equation (12);
5. Compute kernel K_G between two sets of semilocal groups with Equation (11);

3. Experiments

In this section, we present experimental results on recognition tasks using local features and SVM classification,

bridged together by the proposed kernels. In principle, the proposed kernels can work with any pattern analysis algorithm that is able to be "kernelized", i.e., depending on data through their scalar products. SVM was chosen for its performance and generalization ability.

3.1. Experimental setup

We performed our experiments on the COIL-100 database [13], a standard test benchmark for object recognition. The COIL-100 database contains 7200 color images of 100 different objects. All images are 128×128 pixels in size. They were obtained by placing the objects on a turntable and taking a picture every 5° of viewing angle of a 360° rotation.

In our experiments, the training set of all SVM classifiers consisted of 3600 images, 36 for each of the 100 objects that correspond to a 10° difference in the viewing angles. Shown in the top row of Figure 2 are five images from the training set. From the remaining images, five different testing sets were formed:

- **Set 1:** 3600 images with viewing angles other than those used in the training set.
- **Set 2:** 3600 images generated by randomly scaling, rotating and translating images in set 1.
- **Set 3:** 3600 images generated by adding Gaussian noise of average 12 dB to the images in set 1.
- **Set 4:** 3600 images generated by embedding the images in set 1 into randomly chosen backgrounds.[5]
- **Set 5:** 3600 images generated by artificially adding partial occlusions (stripes from randomly chosen images) to the images in set 1.

Set 1 and 2 test the generalization ability of the kernels and classifiers to changes in viewing angles and object positions. Set 3-5 are devised to test their resilience to common image degradations, namely additive noise, background clutters and partial occlusions.

On each image, three types of local features along with their corresponding minor kernels were computed:

1. **Local jets** [17] are differential grayvalue invariants computed around an interest point. Each local jet is a vector of dimension 9 containing up to the third order derivatives. A Mercer kernel between two local jet features, x and z, is $K(x,z) = \exp\left(-\frac{(x-z)^T\Lambda^{-1}(x-z)}{2\sigma^2}\right)$, where Λ is the covariance matrix and $\sqrt{(x-z)^T\Lambda^{-1}(x-z)}$ is the Mahalanobis distance between x and z.

[5]Images used for backgrounds and partial occlusions in set 4 and set 5 are downloaded from `http://www.freefoto.com`.

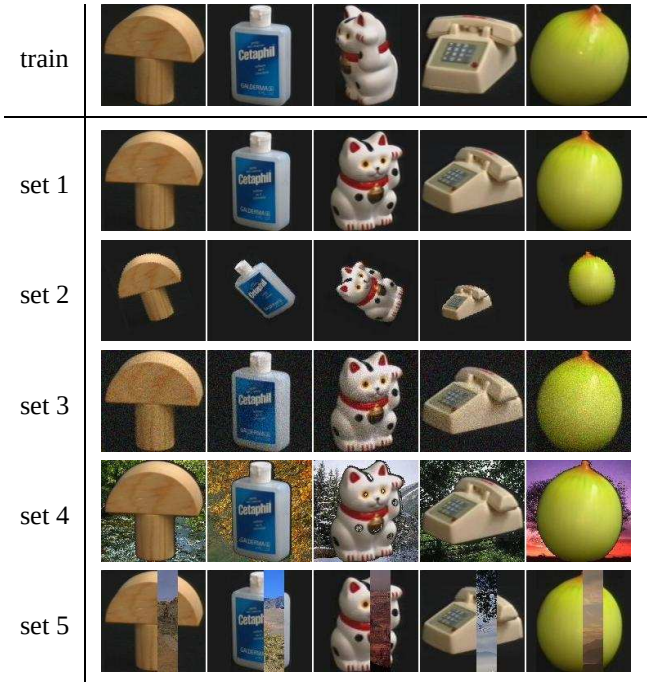

Figure 2: Examples of images used in our experiments. The first row are images from the training set. The remaining rows are examples from each of the five testing sets.

2. **Local histograms** [16] are local features consisting of histogram at different scales around interest points. Using 32 bins in computing the histogram and considering up to 3 scales, each feature is a 96 dimensional vector. A kernel based on the χ^2-similarity between two feature vectors, x and z, $K(x,z) = \exp\left(-\frac{\chi^2(x,z)}{2\sigma^2}\right)$, is introduced in [22] and proved to satisfy the Mercer condition [4].
3. **Local phase-based features** [2] are comprised by local phases of a complex pyramid decomposition of the image. The features are 36-dimensional complex-valued vectors, and their similarity is measured by $\mathcal{C}(x,z) = \left|\frac{xz^*}{1+|x||z|}\right|$, from which a Mercer kernel is constructed as $K(x,z) = (\mathcal{C}(x,z)+1)^q$.

For each of these local features, interest points were found by a Harris corner detector, showed to have high repeatability and robust performance [18]. Interest points too close to the boundary were ignored to avoid image border effects. The parameters of the interest point detector were set so that, on average, approximately 100 interest points were found in an image. Semilocal groups, as described in section 2.6, were formed on each interest point using its five nearest neighbors.

To have a basis of comparison, we also collected a global

feature from each image. The global feature we used is the raw pixel representation [15], which was obtained by first converting a 128×128 color image into grayscale and resizing it to 32×32 pixels. A 1024-dimensional feature vector was formed by stacking the grayvalues of the resized image.

For the local feature representations, composite kernels as described in Section 2.8 were formed from the local features and their kernels. The kernel parameter, p, was set to 9 in all cases. For the global features, a Gaussian kernel, Equation (2) was employed. The SVM classifiers were implemented with package LIBSVM [3], which was enhanced to work with kernels on local feature representations. As a standard preprocessing step in the literature, we used the normalized kernel evaluation in building the SVM classifier, as $K(x, y) \leftarrow \frac{K(x,y)}{\sqrt{K(x,x)}\sqrt{K(y,y)}}$. We employed a simple multi-class protocol for classification, namely a one-versus-the-rest scheme in training and a winner-takes-all strategy in testing. The regularization parameter of SVM was set to 10^3 in all classifiers.

3.2. Results

Shown in Table 1 is the performance of different types of kernels with the local jets on all testing sets. For comparison, the performance of the global feature (raw pixel representation) with a Gaussian kernel is also included. Performance is evaluated in error rates, which is the percentage of all misclassification cases in all testing examples. Several points are worth noting about this set of results. First, the local jet features out-performed the global features on all testing sets. The difference is more significant with the presence of image transformations and degradations (set 2-5). Furthermore, notice that the proposed kernel, $K_{\mathcal{F}}$, achieved competitive performance to that of the matching kernel K_M. As K_M is less sensitive to mismatches in local features, it had lowest error rate in some cases (set 3,4). However, its drawback is that there is no guarantee of a unique global optimal solution to the SVM training.

Shown in Figure 3 is a plot of the contribution of the best matched local feature pairs in the evaluation of kernel $K_{\mathcal{F}}$, Equation (8), with regards to the kernel parameter p. For stability, we reported here the average over kernel evaluations of all training image pairs. Notice that after $p \geq 9$, this ratio is plateaued to be more than 99%, indicating that the best matched pairs of local features has dominated in the kernel evaluation. This fact is further supported by the corresponding classification error rates of $K_{\mathcal{F}}$ on test set 1, Figure 4. With p chosen greater than 9, the performance does not improved significantly.

In the second series of experiments, we tested the proposed kernel combined with different types of local features. Shown in Table 2 are the results of this experiment. Note that the local jets work well under noise, but suffer from background clutter and occlusion. The local histograms, on the other hand, are more robust in the face of partial occlusions. The local phase-based features perform worst in all the experiments.

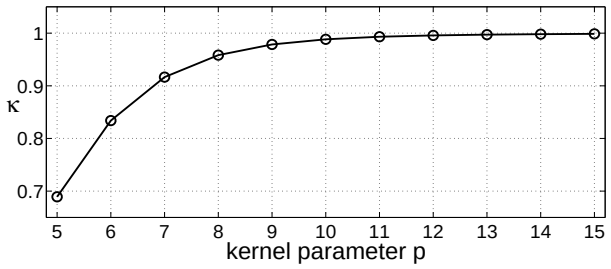

Figure 3: Contribution of the best matched local feature pairs, κ, Equation (8), in kernel $K_{\mathcal{F}}$ with local jet features as a function of the kernel parameter p.

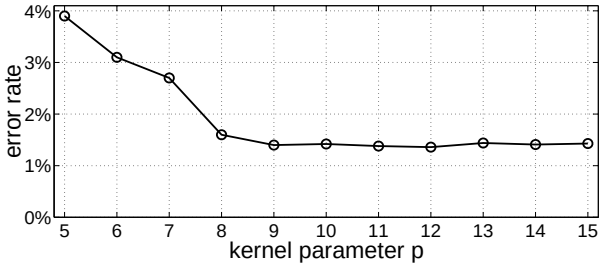

Figure 4: Classification performance of kernel $K_{\mathcal{F}}$ with local jet features on set 1 as a function of the kernel parameter p.

We further combined all types of local features as in Equation (10), and reported its performance in the first row of Table 3. It seems that fusion of local features does not necessarily improve the performance (set 1). However, in cases of image degradations, this approach achieved better results, possibly because multiple types of local features provide complementary information that helps to reduce ambiguity in classification. Finally, we constructed an SVM classifier using the kernel defined in Equation (11), to further incorporate the semilocal constraints. Such a kernel, equipped with the most comprehensive domain knowledge, was expected to work best. Shown in the second row of Table 3 is the performance of this kernel on all testing sets. Compared to other kernels, it indeed achieved the lowest error rate, which suggests the efficacy of semilocal constraints.

4. Discussion

In this paper, we have introduced a new class of kernels for appearance based object recognition with local feature rep-

	set 1	set 2	set 3	set 4	set 5
K_R	11.3	53.9	38.4	57.3	49.6
K_S	10.9	29.4	23.9	39.7	36.4
K_M	1.8	27.2	**16.9**	**20.5**	28.8
$K_{\mathcal{F}}$	**1.4**	**23.8**	17.7	25.3	**24.6**

Table 1: Error rates (in percentage) of the Gaussian kernel (Equation (2)) with a global feature (raw pixels) and different kernels, K_R (Equation (3)),K_S (Equation (4)) and K_M (Equation (5)), with the local jet features.

	set 1	set 2	set 3	set 4	set 5
local jets	**1.4**	**23.8**	**17.7**	**25.3**	24.6
local histograms	7.6	34.1	22.8	28.6	**21.2**
local phases	10.2	39.4	28.5	29.1	27.9

Table 2: Error rates (in percentage) of different local features with kernel $K_{\mathcal{F}}$, Equation (5).

	set 1	set 2	set 3	set 4	set 5
K_F	6.1	23.5	14.4	19.9	21.2
K_G	1.2	17.6	8.3	12.7	10.9

Table 3: Error rates (in percentage) of kernels using multiple types of local features, Equation (10) and with semilocal constraints, K_G, Equation (11).

resentations. The proposed kernels work on sets of local features with variable lengths, satisfy the Mercer condition and reflect similarities between sets of local features. In addition, multiple types of local features and semilocal constraints were combined into the kernel design, which help to further improve the performance. We presented preliminary experimental results where the proposed kernels, coupled with SVM classification, showed promising performance in recognition tasks and is robust to image transformations and degradations.

Acknowledgment

I thank Hany Farid for helpful and inspiring comments about the paper. This work was supported by Hany Farid under an Alfred P. Sloan Fellowship, an NSF CAREER Award (IIS99-83806), an NSF Infrastructure Grant (EIA-98-02068), and under Award No. 2000-DT-CX-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security (points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security).

Appendix A: K_M is not a Mercer kernel

Proposition 5 *Function K_M defined in Equation (3) is not a Mercer kernel, given that K_f is a Mercer kernel defined on the local features.*

Proof To prove that K_M is not a Mercer kernel, according to Proposition 1, it is sufficient to show that there is a subset of the input space on which the matrix evaluated with K_M is not positive semi-definite.

To this end, consider $\mathcal{F}_1 = \{f_1, f_2\}$, $\mathcal{F}_2 = \{f_3, f_4\}$, and $\mathcal{F}_3 = \{f_5, f_6\}$. Assume a kernel matrix on set $\{f_1, \cdots, f_6\}$ constructed by some kernel function as

$$G_f = \begin{pmatrix} 127 & 127 & 141 & 60 & 159 & 128 \\ 127 & 287 & 215 & 127 & 236 & 135 \\ 141 & 215 & 206 & 101 & 223 & 157 \\ 60 & 127 & 101 & 73 & 134 & 79 \\ 159 & 236 & 223 & 134 & 281 & 191 \\ 128 & 135 & 157 & 79 & 191 & 160 \end{pmatrix}.$$

Matrix G_f is positive semi-definite, evident from its singular value decomposition. From G_f, the kernel matrix of K_M can be constructed using Equation (3) as

$$G_M = \begin{pmatrix} 207.00 & 174.50 & 191.50 \\ 174.50 & 153.50 & 184.25 \\ 191.50 & 184.25 & 236.00 \end{pmatrix}.$$

For example, the element of G_M at row 1 and column 3 is computed as

$$\begin{aligned}(G_M)_{13} &= \frac{1}{4}\left[\max((G_f)_{15}, (G_f)_{16}) + \max((G_f)_{25}, (G_f)_{26})\right. \\ &+ \left.\max((G_f)_{51}, (G_f)_{52}) + \max((G_f)_{61}, (G_f)_{62})\right] \\ &= (159 + 236 + 236 + 135)/4 = 191.50.\end{aligned}$$

Matrix G_M is not positive semi-definite, as its eigenvalues are -1.77, 29.61, and 568.65. Therefore, using K_M can not always form positive semi-definite matrix and this means K_M is not a Mercer kernel.

Appendix B: Proof of Equation (8)

Without loss of generality, let us assume $K_F(F_i^{(a)}, F_1^{(b)}) \geq \cdots \geq K_F(F_i^{(a)}, F_{|\mathcal{F}_b|}^{(b)})$ and $F_1^{(b)})$ is the unique best match feature to $K_F(F_i^{(a)}$ in $\mathcal{F}_b$[6]. The contribution of the best matched local feature in $\mathcal{F}_b$ with $F_i^{(a)}$ in the sum of Equation (6) is:

$$\kappa = \left[K_F(F_i^{(a)}, F_1^{(b)})\right]^p / \sum_{j=1}^{|\mathcal{F}_b|}\left[K_F(F_i^{(a)}, F_j^{(b)})\right]^p. \quad \text{(b.1)}$$

[6]This constraint is held for most of the cases, but when multiple best matches exist, a similar result can be obtained in the same way.

We require that the similarity of the best matched pair in the sum has a fraction above a given threshold ρ as:

$$\left[K_F(F_i^{(a)}, F_1^{(b)})\right]^p / \sum_{j=1}^{|\mathcal{F}_b|} \left[K_F(F_i^{(a)}, F_j^{(b)})\right]^p \geq \rho. \quad \text{(b.2)}$$

Note that

$$\left[K_F(F_i^{(a)}, F_1^{(b)})\right]^p + (|\mathcal{F}_b| - 1)\left[K_F(F_i^{(a)}, F_2^{(b)})\right]^p \geq \sum_{j=1}^{|\mathcal{F}_b|} \left[K_F(F_i^{(a)}, F_j^{(b)})\right]^p. \quad \text{(b.3)}$$

Therefore,

$$\frac{\left[K_F(F_i^{(a)}, F_1^{(b)})\right]^p}{\left[K_F(F_i^{(a)}, F_1^{(b)})\right]^p + (|\mathcal{F}_b| - 1)\left[K_F(F_i^{(a)}, F_2^{(b)})\right]^p} \geq \frac{\left[K_F(F_i^{(a)}, F_1^{(b)})\right]^p}{\sum_{j=1}^{|\mathcal{F}_b|} \left[K_F(F_i^{(a)}, F_j^{(b)})\right]^p} \geq \rho,$$

Rearranging terms yields

$$p \geq \log \frac{1-\rho}{(|\mathcal{F}_b| - 1)\rho} / \log \frac{K_F\left(F_i^{(a)}, F_2^{(b)}\right)}{K_F\left(F_i^{(a)}, F_1^{(b)}\right)}. \quad \text{(b.4)}$$

Appendix C: Proof of Proposition 4

Proof Consider two n-dimensional vectors $x = (x_0, \cdots, x_{n-1})^T$ and $y = (y_0, \cdots, y_{n-1})^T$, define X and Y to be subsets of $\mathcal{R}^n$ as $X = \{c(x,0), \cdots, c(x, n-1)\}$ and $Y = \{c(y,0), \cdots, c(y, n-1)\}$, where $c : \mathcal{R}^n \times \{0, \cdots, n-1\} \to \mathcal{R}^n$ is the circular-shift operator in $\mathcal{R}^n$. Now define

$$\widetilde{K}(X,Y) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} [K(c(x,i), c(y,j))]^p. \quad \text{(c.1)}$$

With a similar proof as that of Proposition 3, we conclude that $\widetilde{K}(\cdot,\cdot)$ is a Mercer kernel. With the definition of Mercer kernels, this suggests that there exists a mapping $\phi_2 : 2^{\mathcal{R}^n} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space, such that

$$\widetilde{K}(\widetilde{X}, \widetilde{Y}) = \langle \phi_2(\widetilde{X}), \phi_2(\widetilde{Y}) \rangle. \quad \text{(c.2)}$$

Expanding the evaluation of kernel $\widetilde{K}$ as

$$\begin{aligned}
&\widetilde{K}(X,Y) = \textstyle\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} [K(c(x,i), c(y,j))]^p \\
&= \textstyle\sum_{i=0}^{n-1} [K(x,y)]^p + \sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} [K(x, c(y, j-i))]^p \\
&+ \textstyle\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} [K(c(x, j-i), y)]^p \\
&= nK(x,y)^p + \textstyle\sum_{i=0}^{n-2}\sum_{l=1}^{n-i-1} [K(x, c(y,l))]^p + \\
&\quad \textstyle\sum_{i=0}^{n-2}\sum_{l=n-i-1}^{n-1} [K(x, c(y,l))]^p \\
&= nK(x,y)^p + \textstyle\sum_{i=0}^{n-2} [K(x, c(y, n-i-1))]^p + \\
&\quad \textstyle\sum_{i=0}^{n-2}\sum_{l=1}^{n-1} [K(x, c(y,l))]^p \\
&= nK(x,y)^p + \textstyle\sum_{l=1}^{n-1} [K(x, c(y,l))]^p + \\
&\quad (n-1)\textstyle\sum_{l=1}^{n-1} [K(x, c(y,l))]^p \\
&= n\textstyle\sum_{l=0}^{n-1} [K(x, c(y,l))]^p = nK_{\mathcal{G}}(x,y).
\end{aligned}$$

Therefore, we have $K_{\mathcal{G}}(x,y) = \frac{1}{n}\widetilde{K}(X,Y)$. Notice that in proving this equality, we used a property of the circular shift operator as $x = c(x,n)$ and our assumption about the base kernel, i.e., $K(x,y) = K(c(x,d), c(y,d))$ for $0 \leq d \leq n-1$.

We can now define another mapping $\phi_1 : \mathcal{R}^n \to 2^{\mathcal{R}^n}$ such that $\phi_1(x) = \{c(x,0), \cdots, c(x, n-1)\}$. Substituting the definitions of mapping ϕ_1 and ϕ_2 into the evaluation of $K_{\mathcal{G}}$ yields

$$\begin{aligned}
K_{\mathcal{G}}(x,y) &= \frac{1}{n}\widetilde{K}(X,Y) = \frac{1}{n}\langle \phi_2(X), \phi_2(Y) \rangle \\
&= \frac{1}{n}\langle \phi_2(\phi_1(x)), \phi_2(\phi_1(y)) \rangle \\
&= \left\langle \frac{1}{\sqrt{n}}\phi_2(\phi_1(x)), \frac{1}{\sqrt{n}}\phi_2(\phi_1(y)) \right\rangle.
\end{aligned}$$

If a new mapping $\phi : \mathcal{R}^n \to \mathcal{H}$ is defined as $\phi(x) = \frac{1}{\sqrt{n}}\phi_2(\phi_1(x))$, we then have

$$K_{\mathcal{G}}(x,y) = \langle \phi(x), \phi(y) \rangle, \quad \text{(c.3)}$$

which shows that $K_{\mathcal{G}}(\cdot,\cdot)$ is a Mercer kernel and hence proves the first part of Proposition 4.

The second part of Proposition 4 can be proved by first noticing that $\phi_1(x) = \{c(x,0), \cdots, c(x, n-1)\} = \phi_1(c(x,d))$ for any $0 \leq d \leq n-1$. Therefore,

$$\begin{aligned}
K_{\mathcal{G}}(x,y) &= \frac{1}{n}\widetilde{K}(\phi_1(x), \phi_1(y)) \\
&= \frac{1}{n}\widetilde{K}(\phi_1(c(x,d_1))\phi_1(c(y,d_2))) \\
&= K_{\mathcal{G}}(c(x,d_1), c(y,d_2)).
\end{aligned}$$

for any given $0 \leq d_1, d_2 \leq n-1$.

References

[1] S. Boughorbel, J. Tarel, and F. Fleuret. Non-mercer kernel for SVM object recognition. In *British Machine Vision Conference (BMVC)*, 2004.

[2] G. Carneiro and A. Jepson. Phase-based local features. In *European Conference on Computer Vision (ECCV)*, 2001.

[3] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] O. Chapelle, P. Haffner, and V. Vapnik. SVMs for histogram based image classifi cation. *IEEE Transactions on Neural Networks*, 10(5), 1999.

[5] J. Eichhorn and O. Chapelle. Object categorization with SVM: Kernels for local features. In *Advances in Neural Information Processing Systems (NIPS)*, page to appear, 2004.

[6] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[7] D. Haussler. Convolution kernel for structure data. Technical Report UCS-CRL-99-10, 1999.

[8] R. Kondor and T. Jebra. A kernel between sets of vectors. In *International Conference on Machine Learning (ICML)*, 2003.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *Internation Journal on Computer Vision*, 60(2):91–110, 2004.

[10] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. *Internation Journal on Computer Vision*, pages 523–531, 2001.

[11] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[12] P. Monreno, P. Ho, and N. Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classifi cation in multimedia applications. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[13] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100). Technical Report CUCS-006-96, Columbia University, 1996.

[14] F. Odone, A. Barla, and A Verri. Building kernels from binary strings for image matching. *IEEE Transaction on Image Processing*, (to appear).

[15] M. Pontil and A. Verri. Support vector machines for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.

[16] F. Schaffalitzky and A. Zissermann. Viewpoint invariant texture matching and wide baseline stereo. In *IEEE International Conference on Computer Vision (ICCV)*, 2001.

[17] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[18] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal on Computer Vision*, 7(1):11–32, 2000.

[19] N. Sebe, Q. Tian, E. Loupias, M. Lew, and T. Huang. Evaluation of salient point techniques. In *International Conference on Image and Video Retrieval*, 2004.

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

[21] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[22] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *IEEE International Conference on Computer Vision (ICCV)*, pages 257–264, 2003.

[23] L. Wolf and A. Shashua. Kernel principle angles for classifi ication machines with applications to image sequence interpretation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.