

Recognition with Local Features: the Kernel Recipe

Christian Wallraven
MPI for Biological Cybernetics
Tübingen, Germany
christian.wallraven@tuebingen.mpg.de

Barbara Caputo
CVAP, NADA, KTH
SE-100 44 Stockholm, Sweden
caputo@nada.kth.se

Arnulf Graf
MPI for Biological Cybernetics
Tübingen, Germany
arnulf.graf@tuebingen.mpg.de

Abstract

Recent developments in computer vision have shown that local features can provide efficient representations suitable for robust object recognition. Support Vector Machines have been established as powerful learning algorithms with good generalization capabilities. In this paper, we combine these two approaches and propose a general kernel method for recognition with local features. We show that the proposed kernel satisfies the Mercer condition and that it is suitable for many established local feature frameworks. Large-scale recognition results are presented on three different databases, which demonstrate that SVMs with the proposed kernel perform better than standard matching techniques on local features. In addition, experiments on noisy and occluded images show that local feature representations significantly outperform global approaches.

1. Introduction

Developing computer vision systems capable of unconstrained recognition of objects has been at the heart of computer vision research for the last decades. Changes in illumination, size and pose, occlusion by other objects and nonrigid deformations are among some of the problems that such a system has to face in real-world conditions; often several of these changes occur at the same time.

Recent years have seen impressive improvements in object recognition performance under such conditions [3, 19], and it seems that appearance-based methods [21, 22, 6, 3] are gaining popularity over structural methods [15]. In this paper we will focus on appearance-based approaches, where objects are modeled by a set of images and recognition is performed by directly matching the input image to

the model set. This model set could consist of the original images, considered as feature vectors [18, 19, 1], or of features extracted from the original views, such as color [24] or texture [21]. In all cases, the features are considered to be representative of the appearance of the objects to be recognized. Within this research area one can identify two main research lines: the first concerns the object representation, that is, how to extract efficient and effective representations from visual input whereas the second focuses on algorithms to process these representations.

The simplest representation - raw pixel data of input images - can achieve surprisingly high recognition rates [18, 19], but is highly sensitive to all signal changes. Furthermore, storage requirements are extraordinarily high if the system is supposed to recognize more than a few dozen objects. Other global representations like color or derivative histograms [24, 21] are in its original form rather sensitive to occlusions. Local representations [22, 16, 12] address both problems as they consist of a number of localized features in the image. Several successful vision systems that use such local feature representations have been developed, and seem to be able to support high recognition performance under real-world conditions [16, 20, 12].

The second line of research focuses on the algorithms used to process the representations both during learning and recognition. 'Learning-free' algorithms such as nearest neighbor techniques can provide a good baseline for recognition experiments, but often suffer from inferior generalization capabilities in real-world conditions [8, 18]. Support Vector Machines (SVMs, [9, 25]), on the other hand, represent a class of learning algorithms, which are based on a thorough mathematical founding, and - while more computationally expensive than other matching techniques - have shown impressive learning and recognition performance [19, 1, 8, 18]. This performance can even be achieved on

relatively simple raw pixel data, thus demonstrating the generalization capability of SVMs.

However, until now it has not been possible to incorporate ‘the best of both worlds’ in a recognition system, that is, to use local representations as input to SVMs. This is due to the intrinsic structure of both techniques: local representations generally consist of feature vectors of different length, and matching has traditionally required the definition of ad-hoc similarity measures [16, 20, 26]. In other words, similarity measures commonly employed for global representations cannot even be computed for local representations. SVMs on the other hand are large margin classifiers, where the optimal separating surface is defined by a linear combination of scalar products between the view to be classified and some “support vectors”. Thus in its standard formulation local features exclude SVMs as classifiers, and vice-versa.

The contribution of this paper is to solve this dichotomy: we define a new class of kernels which satisfies Mercer condition as well as allowing the computation of scalar products on local features. We show how this class of kernels is related to similarity measures proposed in the literature for some well-known local features. Furthermore, we show through extensive experiments, how local features combined with SVM, via this type of kernel, outperform local features combined with state-of-the-art matching techniques, as well as SVMs with global representations.

The paper is organized as follows: after reviewing previous literature (section 2) and SVMs (section 3), we discuss local features and their difficulties in using them as input to a SVM (section 4). Section 5 introduces the new class of local kernels, demonstrates that they satisfy Mercer condition and shows how they are related to state-of-the-art similarity measures for local features. Section 6 reports and discusses experiments on three different databases, with occluded and cluttered views.

2 Previous Work

Object recognition is one of the most active areas of computer vision with applications in many fields. Many researchers have approached the problem with appearance-based methods. Swain and Ballard [24] proposed to represent an object by its color histogram, which was shown to be robust to changes in orientation, scale, partial occlusion and changes of the viewing position. The major drawbacks of this method are sensitivity to lighting conditions and that for many object classes color is not a discriminative feature. Schiele and Crowley [21] generalized this method by introducing multidimensional receptive field histograms to approximate the probability density function of local appearance. Their recognition algorithm calculates probabilities for the presence of objects based on a small number of

vectors of local neighborhood operators such as Gaussian derivatives at different scales.

Based on local characteristics, Schmid and Mohr [22] developed a system that can recognize objects in the case of partial visibility, image transformations and within complex scenes. Their approach is based on the combination of differential invariants computed at key points with a robust voting algorithm and semi-local constraints. Recognition is based on the computation of the similarity (represented by the Mahalanobis distance) between two invariant vectors. Matching is performed on discriminant points of an image, and a standard voting algorithm is used to find the closest model to an image. The idea was further developed by many authors in order to include invariances (such as viewpoint invariance [20], affine invariance [16], scale-space selection [12]).

Recently, SVMs and kernel methods have begun to be used for appearance-based object recognition. Pontil [18] demonstrated the robustness of SVMs to noise, bias in the registration and moderate amount of partial occlusions. Roobaert et al. [19] examined the generalization capability of SVMs, when just a few number of views per objects are available. Barla et al. [1] proposed to use a new class of kernels, especially designed for vision and inspired by similarity measures successfully employed in other vision applications (including histogram intersection and Hausdorff kernels). The growing number of papers addressing object recognition using kernel methods (see for instance [14, 6, 26] and many others) is an indicator for the interest of the computer vision community in this area. However, a common limitation of all these approaches (with the noticeable exception of [26]) is that they can handle only *global* features.

3 Support Vector Machines

In this section we give a brief overview of binary classification with SVMs. For further details and the extension to multiclass settings we refer the reader to [9, 25]. Consider the problem of separating the set of training data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)$ into two classes, where $\mathbf{x}_i \in \mathbb{R}^N$ is a feature vector and $y_i \in \{-1, +1\}$ its class label. If we assume that the two classes can be separated by a hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$, and that we have no prior knowledge about the data distribution, then the optimal hyperplane (that is, the one with the lowest bound on the expected generalization error) is the one which maximizes the margin [9, 25]. The optimal values for \mathbf{w} and b can be found by solving the following constrained minimization problem:

$$\begin{aligned} & \underset{\mathbf{w}, b}{\text{minimize}} && \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to} && y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \forall i = 1, \dots, m \end{aligned} \quad (1)$$

Solving it using Lagrange multipliers $\alpha_i (i = 1, \dots, m)$ results in a classification function

$$f(x) = \text{sgn} \left(\sum_{i=1}^m \alpha_i y_i \mathbf{w} \cdot \mathbf{x} + b \right). \quad (2)$$

where α_i and b are found by using an SVC learning algorithm [9, 25]. Most of the α_i 's take the value of zero; those \mathbf{x}_i with nonzero α_i are the so-called "support vectors". In cases where the two classes are non-separable, the solution is identical to the separable case with a modification of the Lagrange multipliers to $0 \leq \alpha_i \leq C, i = 1, \dots, m$, where C is the penalty for the misclassification. To obtain a non-linear classifier, one maps the data from the input space \mathcal{R}^N to a high dimensional feature space \mathcal{H} by $\mathbf{x} \rightarrow \Phi(\mathbf{x}) \in \mathcal{H}$, such that the mapped data points of the two classes are linearly separable in the feature space. Assuming there exists a kernel function K such that $K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$, then a nonlinear SVM can be constructed by replacing the inner product $\mathbf{x} \cdot \mathbf{y}$ in the linear SVM by the kernel function $K(\mathbf{x}, \mathbf{y})$

$$f(x) = \text{sgn} \left(\sum_{i=1}^m \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right) \quad (3)$$

This corresponds to constructing an optimal separating hyperplane in the feature space.

4 Support Vector Machines and Local Features

Now we turn to the problem of using SVMs with local features. Given a set of images $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^m$, the most general local feature vector for the image \mathbf{I}_i can be described as $\mathbf{L}_i = \{\mathbf{l}_j(\mathbf{I}_i), \mathbf{p}_j(\mathbf{I}_i)\}_{j=1}^{n_i}$, computed as follows:

- an interest point detector (a popular choice is the Harris corner detector, [22, 20, 12]) detects n_i points. In general, the number of interest points detected for each image \mathbf{I}_i will differ;
- $\mathbf{p}_j(\mathbf{I}_i)$ are the coordinates (in the image plane) of the j -th point;
- $\mathbf{l}_j(\mathbf{I}_i)$ is a feature vector computed locally around the j -th point (see for instance [22, 26, 20]).

When one does not consider interest point coordinates, the local feature vector reduces to $\mathbf{L}_i = \{\mathbf{l}_j(\mathbf{I}_i)\}_{j=1}^{n_i}$.

We see immediately that local features cannot be used in a straightforward way as input for SVMs, as they have different lengths for different images, which makes it impossible to perform scalar products. One way around this might be to simply add an appropriate number of zeros to

each feature vector in order to normalize vector lengths. Let us examine this proposition by considering two local feature vectors $\mathbf{L}_1 = \{\mathbf{l}_j(\mathbf{I}_1)\}_{j=1}^{n_1}$ and $\mathbf{L}_2 = \{\mathbf{l}_j(\mathbf{I}_2)\}_{j=1}^{n_2}$, with $n_2 > n_1$ (the argument can be extended easily to the case of local features including point coordinates). We can define a new feature vector $\tilde{\mathbf{L}}_1$ by zero-padding \mathbf{L}_1 :

$$\tilde{\mathbf{L}}_1 = \underbrace{\{\mathbf{l}_1(\mathbf{I}_1), \dots, \mathbf{l}_{n_1}(\mathbf{I}_2)\}}_{n_1}, \underbrace{\{0, \dots, 0\}}_{n_2 - n_1}$$

This would allow us to compute the scalar product between \mathbf{L}_2 and $\tilde{\mathbf{L}}_1$:

$$\tilde{\mathbf{L}}_1 \cdot \mathbf{L}_2 = \mathbf{l}_1(\mathbf{I}_1) \cdot \mathbf{l}_1(\mathbf{I}_2) + \mathbf{l}_2(\mathbf{I}_1) \cdot \mathbf{l}_2(\mathbf{I}_2) + \dots + \mathbf{l}_{n_1}(\mathbf{I}_1) \cdot \mathbf{l}_{n_1}(\mathbf{I}_2) + 0 \cdot \mathbf{l}_{n_1+1}(\mathbf{I}_2) + \dots + 0 \cdot \mathbf{l}_{n_2}(\mathbf{I}_2)$$

Whereas it is technically possible to compute scalar products for local features using this trick, this computed quantity is of small interest (if any) from the point of view of recognition. This is because the underlying philosophy in describing an image by local features is that, once "interesting points" in the image are detected, local descriptors are computed around these points. Such a local descriptor should be discriminative in the sense that, if the point is detected again in a new image, the comparison of the descriptors computed around the points will allow them to match correctly. Thus one can see that local features are effective for recognition if and only if the algorithm we use measures similarities between *all* local features within the compared images. This is exactly what state-of-the-art algorithms for matching and recognition do (see, e.g., [22, 20, 12]). Coming back to SVMs, this means that the issue is not just to be able to perform scalar products. If we want to benefit from the power of large margin classifiers when using local features, we must turn to different strategies for measuring local similarities with scalar products. In other words, we need to define a new class of kernels.

5. Local Kernels

In this section we define a new class of kernels for local features and prove that they satisfy Mercer's theorem. In addition, we also show how these kernels can be applied to some existing approaches for matching and recognition using local features.

We begin by recalling that a kernel function $K(\mathbf{x}, \mathbf{y})$ must satisfy Mercer's theorem [9, 25], and that it holds that:

Proposition 1 ([9], Proposition 3.5, Chap. 3, pg 33)

Let X be a finite input space with $K(\mathbf{x}, \mathbf{y})$ a symmetric function on X . Then $K(\mathbf{x}, \mathbf{y})$ is a kernel function if and only if the matrix

$$\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{(i,j)=1}^m$$

is positive semi-definite (has non-negative eigenvalues).

Then it holds

Theorem 2 Denote by $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^m$ a set of images and $\mathcal{L} = \{\mathbf{L}_i\}_{i=1}^m$ the corresponding set of local features, with $\mathbf{L}_i = \{\mathbf{l}_j(\mathbf{I}_i)\}_{j=1}^{n_i}, i = 1, \dots, m$. For all $(\mathbf{L}_h, \mathbf{L}_k) \in \mathcal{L}$, consider the function

$$K_L(\mathbf{L}_h, \mathbf{L}_k) = \frac{1}{2} \left[\hat{K}(\mathbf{L}_h, \mathbf{L}_k) + \hat{K}(\mathbf{L}_k, \mathbf{L}_h) \right] \quad (4)$$

with

$$\hat{K}(\mathbf{L}_h, \mathbf{L}_k) = \frac{1}{n_h} \sum_{j_h=1}^{n_h} \max_{j_k=1, \dots, n_k} \{K_l(\mathbf{l}_{j_h}(\mathbf{L}_h), \mathbf{l}_{j_k}(\mathbf{L}_k))\}.$$

If $K_l(\mathbf{l}_{j_h}, \mathbf{l}_{j_k})$ is a Mercer kernel, then $K_L(\mathbf{L}_h, \mathbf{L}_k)$ is a Mercer kernel.

Proof Note first that $K_L(\mathbf{L}_h, \mathbf{L}_k)$ is symmetric by construction. Then, if K_l is a Mercer kernel, it follows by proposition 1 that it is positive semi-definite. Thus, the operation of max will result in a Mercer kernel as well (it does nothing but choose one of the n_k Mercer kernels). This means that equation (4) is a linear combination with positive coefficients of Mercer kernels; it follows that $K_L(\mathbf{L}_h, \mathbf{L}_k)$ is a Mercer kernel. ■

When local features include point coordinates, we can use this additional information in the form of a simple position constraint by extending the local kernel as follows:

Theorem 3 Denote by $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^m$ a set of images and $\mathcal{L} = \{\mathbf{L}_i\}_{i=1}^m$ the corresponding set of local features, with $\mathbf{L}_i = \{\mathbf{l}_j(\mathbf{I}_i), \mathbf{p}_j(\mathbf{I}_i)\}_{j=1}^{n_i}, i = 1, \dots, m$. For all $(\mathbf{L}_h, \mathbf{L}_k) \in \mathcal{L}$, consider the function

$$K_{LP}(\mathbf{L}_h, \mathbf{L}_k) = \frac{1}{2} \left[\hat{K}(\mathbf{L}_h, \mathbf{L}_k) + \hat{K}(\mathbf{L}_k, \mathbf{L}_h) \right] \quad (5)$$

with

$$\hat{K}(\mathbf{L}_h, \mathbf{L}_k) = \frac{1}{n_h} \sum_{j_h=1}^{n_h} \max_{j_k=1, \dots, n_k} \{K_l(\mathbf{l}_{j_h}(\mathbf{L}_h), \mathbf{l}_{j_k}(\mathbf{L}_k)) \cdot \exp\{-(\mathbf{p}_{j_h}(\mathbf{L}_h) - \mathbf{p}_{j_k}(\mathbf{L}_k))^2/2\sigma^2\}\}.$$

If $K_l(\mathbf{l}_{j_h}, \mathbf{l}_{j_k})$ is a Mercer kernel, then $K_{LP}(\mathbf{L}_h, \mathbf{L}_k)$ is a Mercer kernel.

Proof $\exp\{-(\mathbf{p}_{j_h}(\mathbf{L}_h) - \mathbf{p}_{j_k}(\mathbf{L}_k))^2/2\sigma^2\}$ is a Mercer kernel (Gaussian kernel, [9, 25]), thus its product with K_l is still a Mercer kernel. Then the argument proceeds as for theorem 2 ■

The key point for the proof of theorem 2 (and consequently of theorem 3) is the condition that K_l is a Mercer kernel. Below we show three examples of state-of-the-art local

features and corresponding algorithms, which fulfill this requirement. This demonstrates the general applicability of the proposed kernels in computer vision.

Example 1 Jet features [22] are a particularly successful example of local features in the literature. Similarity between jet features, which are differential intensity invariants computed around interest points, is measured via the Mahalanobis distance [22]:

$$d_M(\mathbf{x}, \mathbf{y}) = \sqrt{\langle \mathbf{x} - \mathbf{y} | \Lambda^{-1} | \mathbf{x} - \mathbf{y} \rangle}$$

where Λ is the covariance matrix of the components. d_M can be easily mapped into an Euclidean distance d_E [22]: the covariance matrix is a real symmetric positive semi-definite matrix, which can be decomposed via SVD:

$$\Lambda^{-1} = P^T D P,$$

with P orthogonal and D diagonal. It follows that

$$d_M(\mathbf{x}, \mathbf{y}) = d_E(\sqrt{D}P\mathbf{x}, \sqrt{D}P\mathbf{y}).$$

Thus we can use any of the following kernels as K_l in equation (4):

$$K_p(\mathbf{x}, \mathbf{y}) = \left((\sqrt{D}P\mathbf{x} \cdot \sqrt{D}P\mathbf{y}) + c \right)^p, p \in \mathcal{N}, c \in \mathbb{R}^+$$

$$K_{Gauss}(\mathbf{x}, \mathbf{y}) = \exp\{-\rho d_E(\sqrt{D}P\mathbf{x}, \sqrt{D}P\mathbf{y})\}.$$

Example 2 Schaffalitzky and Zissermann [20] proposed to compute local histograms at different scales around detected points of interest; they compare the local features via χ^2 similarity measures. For these local features one can use as K_l the intersection measure introduced by Swain and Ballard [24], which was proven to be a Mercer kernel [1], or

$$K_{\chi^2}(\mathbf{x}, \mathbf{y}) = \exp\{-\rho \chi^2(\mathbf{x}, \mathbf{y})\}, \quad (6)$$

$$K_{a,b}(\mathbf{x}, \mathbf{y}) = \exp\{-\rho \|\mathbf{x}^a - \mathbf{y}^a\|^b\}, \quad (7)$$

with $a \in \mathbb{R}^+, b \in]0, 2]$. Both are Mercer kernels [2, 25], and both have been successfully used with histogram features [6, 8].

Example 3 In [26] a first application of local SVM kernels was given, which were used on tracked local features. The kernel used was similar to \hat{K} in equation (4), with K_l given by

$$K_l = \exp\left\{-\rho \left(1 - \frac{\langle \mathbf{x} - \boldsymbol{\mu}_x | \mathbf{y} - \boldsymbol{\mu}_y \rangle}{\|\mathbf{x} - \boldsymbol{\mu}_x\| \|\mathbf{y} - \boldsymbol{\mu}_y\|}\right)\right\} \quad (8)$$

which satisfies Mercer condition.



Figure 1. Exemplars from the COIL (top row), COGVIS-ETH (middle row) and FACE (bottom row) databases. Note the different degrees of homogeneity of the object classes.

6. Experiments

This section presents recognition experiments showing that SVM and local features, combined together via our local kernel, outperform recognition techniques widely used in computer vision literature. We ran experiments on three different databases, and with three different feature types (two global and one local). Our databases vary in homogeneity and types of object classes used, but all contain rotations of 3D objects, which enabled us to study the degree of view generalization of the chosen recognition methods. This task is especially well-suited to examine the performance of the classifiers in real-world conditions, as viewpoint rotations introduce non-trivial changes in the image. For each experiment, SVMs were benchmarked against a nearest neighbor classifier (NNC). For each feature type, we chose an appropriate similarity measure for both classifiers with the aim of enabling a fair comparison between all conditions.

In the following we describe in detail the experimental settings (section 6.1). Section 6.2 describes and discusses recognition results on the three databases, using all feature types, for both classifiers. Section 6.3 describes and discusses recognition results in presence of noise and occlusion, for one database, all feature types and both classifiers.

6.1 Experimental Setup

6.1.1 Databases

The COIL database ([17], Figure 1, top row) is one of the best known benchmarks for object recognition algorithms. It consists of 7200 color images of 100 objects (72 views for object); each image is 128×128 pixels. The images were obtained by placing the objects on a turntable and taking a view every 5° . Our training set consisted of a subset of 17 views per object, resulting in a view every 20° .

The COGVIS-ETH database ([13], Figure 1, middle row) is a recently released database, consisting of 80 objects from 8 different categories (apple, tomato, pear, toy-cows, toy-horses, toy-dogs, toy-cars and cups). Each object is represented by 41 images from viewpoints spaced equally over the upper viewing hemisphere, at distances of $22.5 - 26^\circ$. Objects are shown on a blue background without rescaling. For training we used a subset of the available views, that is, 16 views per object, spaced 22.5° .

The FACE database ([4], Figure 1, bottom row) consists of 100 faces (50 male, 50 female). Each image is a high-quality computer graphic rendering of a laser-scanned face. Face images are re-sized and color-equalized in order to avoid scanning artifacts. The dataset consists, for each face, of 13 views spaced 15 degrees from left to right profile view; faces are rendered on a black background.

The test set for each database was chosen so that its views were *in between* training views. With this experimental procedure, classifiers have to recognize objects under a rotation of 15 degrees for the FACE, 20 degrees for the Columbia and 22.5 degrees for the COGVIS database, respectively. Given the size and complexity of the databases, this represents a hard recognition problem for any classification scheme.

6.1.2 Image representations

For the experiments we used two global representations (raw pixels and color histograms) and one local representation (differential invariants) (Figure 2). Raw pixel representations were extracted from the databases by conversion of all images to 32×32 pixel grey-level images, thus resulting in a vector with 1024 dimensions. Color histograms were evaluated on the full-size color images with 10 bins in each of the three color channels (R,G and B) and normalized to the bin with the highest pixel counts. As local representation, we chose the jet features proposed in [22],

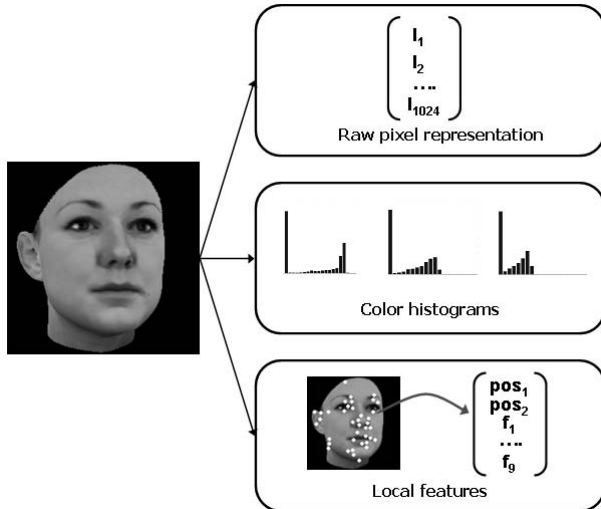


Figure 2. The three representations used in the experiments.

which consist of a 9-dimensional vector computed around a number of interest points over several scales. Detection of interest points was done using a standard Harris-type corner detector, which was shown to have high repeatability and robust performance [23]. On average, such a representation contained around 100 features per image.

6.1.3 Classifiers

For each database and feature type, we ran experiments with NNC and SVM. One might wonder whether this comparison is fair: it is certainly possible to use more sophisticated methods for classification than just simple NNC, such as voting schemes and decision trees [10, 22, 21]. However, our main reason to use NNC is that all of these more advanced methods are used on the basis of a basic classifier, and that most of the time this classifier consists of a NNC. In addition, it should be possible to incorporate such classification methods also into the SVM classification protocol.

All SVM experiments were ran using the SVMlight software [11], where we added our local kernels to the kernel library. In all experiments, ρ was selected via cross-validation on the test set. Since our experiments require a multi-class protocol for classification, we implemented a 1-versus-the-rest scheme for training and a winner-takes-all strategy for testing (see, e.g., [25]).

Distance metrics for NNC, and accordingly kernel functions for SVMs, varied with representations. For the raw pixel representation, we used a standard Euclidean distance, that is, $d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\langle \mathbf{x} - \mathbf{y} | \mathbf{x} - \mathbf{y} \rangle}$. The corresponding kernel (in the sense that it is the kernel which maps the data in an Euclidean space [5]) is the Gaussian kernel (7), with

$$a = 1, b = 2^{-1}.$$

For color histograms we used the standard χ^2 distance, that is, $d_{\chi^2}(\mathbf{x}, \mathbf{y}) = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}$, where the corresponding kernel is the Gaussian kernel (6). Finally, we used the following distance metric for comparing local features:

$$d_L(\mathbf{x}, \mathbf{y}) = \sum_i \max_{j=1, \dots, n_k} \frac{\langle \mathbf{x} - \mu_{\mathbf{x}} | \mathbf{y} - \mu_{\mathbf{y}} \rangle}{\|\mathbf{x} - \mu_{\mathbf{x}}\| \|\mathbf{y} - \mu_{\mathbf{y}}\|}.$$

Note that this metric does not make use of information contained in the feature positions, such as local feature constellations [22], or global feature layout [26]. We have chosen this simpler approach in order to examine the usefulness of the local features themselves. The corresponding kernel is given by (8), with $K_l = \frac{\langle \mathbf{x} - \mu_{\mathbf{x}} | \mathbf{y} - \mu_{\mathbf{y}} \rangle}{\|\mathbf{x} - \mu_{\mathbf{x}}\| \|\mathbf{y} - \mu_{\mathbf{y}}\|}$.

6.2 Experimental Results: Uncorrupted Views

Table 1 reports error rates (e.r.) for all databases, all feature types and both classifiers. From the analysis of these results, we draw three main conclusions: (1) regardless of the data representation, SVMs show large performance improvements compared to NNCs ranging from a minimum of -1.2 % e.r. for the FACE database (local features) to a maximum of -54.2 % e.r. also on the FACE database (raw pixels). As the experimental setup is identical for both classifiers, this large improvement provides further evidence for the superior generalization properties of SVMs. (2) Using SVMs with our local kernel, we achieve the best performance on all databases. (3) On average, the local representation outperforms both global representations. In the following, we discuss results in more detail:

Raw pixels: NNC performance on the COIL and COGVIS-ETH database seems reasonably good (16.7% and 18.9% e.r.) considering the simplicity of both representation and classifier. However, the homogeneous face database shows far higher classification errors (65% e.r.) due to many false matches across views.

Color histograms: Given the low dimensionality of the histograms, recognition results are surprisingly good (around 6% e.r.) for both the FACE and COIL database. We believe that the two main reasons for the inferior performance on the COGVIS-ETH database (23.5% e.r.) are that the blue background dominates the histogram, and that objects *within* categories look very similar thus resulting in more false matches (see Figure 1 middle row showing two similar looking apples).

Local features: With this representation, NNC gives better results than on global features (with the exception of color histograms for the COIL database). Performance on

¹We have chosen Gaussian kernels over polynomial kernels as there is experimental evidence that Gaussian kernels perform better for object recognition tasks [7].

Algorithm	COIL	COGVIS-ETH	FACE
Raw pixel representation			
NNC	16.7%	18.9%	65.0%
SVM	7.2%	10.2%	10.8%
Color histogram representation			
NNC	5.9%	23.5%	6.8%
SVM	3.5%	5.9%	2.7%
Jet Feature representation			
NNC	11.7%	18.5%	1.2%
SVM	1.5%	1.6%	0.0%

Table 1. Classification errors on three databases for global and local representations, using SVM and NNC.

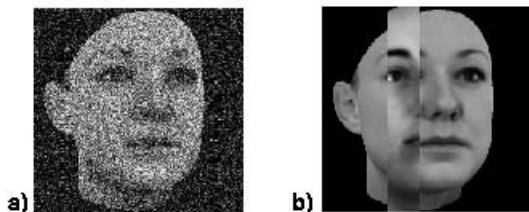


Figure 3. Two types of image degradation: a) Gaussian Noise and b) Occlusion

the FACE database is especially good (1.2% e.r.), which is due to the large amount of discriminative features in the image. This is in contrast to the result on the COGVIS-ETH database (18.5% e.r.), where variation in object size leads to varying numbers of discriminative features.

6.3 Experimental Results: Corrupted Views

We tested robustness to noise and occlusion of our kernel, by adding these two types of image degradation to the *test set* of the FACE database. We limited ourselves to this database because here SVM and NNC yield very similar performance on local features (see Table 1, last column, bottom). Gaussian noise of 10% strength was added to each image² (Figure 3a), which is a manipulation of the global statistics of the image. Occlusion consisted of masking out a random part of the image by inserting data from a different image (Figure 3b), which represents a more local disruption of image statistics. The portion of the face that was masked was set to 15% of the image size. The task for the classifier is thus to recognize objects *both* under depth rotations and additional noise or occlusion.

Results are shown in Table 2. One can see that recognition performance has decreased for all representations

²Note, that this applies to all three color channels.

Algorithm	Noise	Occlusion
Raw pixel representation		
NNC	76.7%	89.1%
SVM	35.0%	58.2%
Color histogram representation		
NNC	98.7%	68.6%
SVM	98.5%	53.2%
Jet Feature representation		
NNC	22.4%	38.2%
SVM	5.4%	26.7%
Jet Feature representation & position constr.		
NNC	9.5%	34.2%
SVM	1.4%	13.2%

Table 2. Classification errors on the FACE database in presence of noise or occlusion.

and classifiers; but once again SVM performed better than NNC, for both kinds of degradation and all feature types. With respect to local features, our results thus confirm their improved robustness to noise and occlusion (Table 2, lower two parts).

Raw pixels: results of both classifiers reasserts that this representation is not robust in presence of noise or occlusion.

Color Histograms: adding Gaussian Noise severely disrupts color information in all three channels, which leads to an extremely poor performance for both classifiers (Table 2, left column, middle). As already observed in [24], color histograms seem to be relatively more robust to occlusion (Table 2, right column, middle). However, performance still drops about 50-60% compared to the uncluttered condition (see Table 1, right column, middle).

Local Features: Local features performs quite well under noise, but suffer from occlusion (although much less than global features). Again, SVM with our local kernel significantly outperforms NNC in both conditions.

Preliminary results on a different kernel (and corresponding metric), which includes a global position constraint in the form of eq. 5 (see also [26]), show that it is possible to significantly improve recognition performance on degraded images by incorporating this extra information. Results for the local representation on the face database (Table 2, last row) show that, by using a position constraint, improvements of up to 13% for both NNC and SVM are possible. Recognition performance is still much better under noise than under occlusion; we believe that here we should be able to perform better by introducing local position constraints (as done in [22] for instance).

7. Conclusion and Outlook

In this paper, we proposed a recipe for constructing kernels which are suitable for object recognition with local features. We showed that these kernels satisfy Mercer condi-

tion, which for the first time opens the possibility to use local features as input for a SVM. We also gave examples of local kernels suitable for several types of local features discussed in the literature. We believe that these types of kernels can be useful for a wide range of applications in the computer vision community. In addition, we presented experiments on three different databases, comparing global versus local features, using NNC and SVM. In all cases, SVM gave significant increases in performance, which again confirms the advantage of large-margin classifiers regardless of the underlying data representation. Moreover, recognition results obtained using local features combined with SVM, via our kernel, outperform recognition results obtained using NNC with local features, as well as SVMs with global representations.

Future work will concentrate on three main directions:

Local position constraint: We will extend our current approach for handling local position information to also yield invariance to affine transformations. Moreover, we will include further local or semi-local position constraints (such as in [22]).

Cue integration: Kernel methods can be a powerful method for cue integration [6]. Our local kernels open the possibility to use this approach also for local features with multiple cues. We expect that this will further increase the recognition performance of our algorithm.

Real world scenarios: We plan to use the proposed kernels for object recognition tasks in real world scenarios, such as recognizing objects in cluttered scenes and under varying scales and illumination conditions. In addition to integrating the results from the previous points into our recognition framework we will also conduct a more detailed evaluation of other local feature representations (see, e.g., [12]).

References

- [1] A. Barla, F. Odone, A. Verri, "Image kernels", *Proc. of ICPR workshop on SVM*, 2002.
- [2] S. Belongie, J. Malik, J. Puchiza, "Matching Shapes", *Proc. ICCV'01*, 2001.
- [3] H. Bishof, H. Wildenauer, A. Leonardis, "Illumination insensitive eigenspaces", *Proc. of ICCV'01*, 2001.
- [4] V. Blanz, T. Vetter, "A Morphable Model for the Synthesis of 3D Faces", *Proc. of SIGGRAPH'99*, 1999.
- [5] C. Burges, "Geometry and Invariance in Kernel based Methods", *Advances in Kernel Methods*, MIT press, 1998.
- [6] B. Caputo, S. Bouattour, H. Niemann, "Robust appearance-based object recognition using a fully connected Markov random field", *Proc. of ICPR'02*, 2002.
- [7] B. Caputo, G. Dorko, "How to combine color and shape information for 3D object recognition: kernels do the trick". *Advances in Neural information processing systems*, Vol 15, 2003.
- [8] O. Chapelle, P. Haffner, and V. Vapnik. SVMs for histogram-based image classification. *IEEE Trans. on Neural Networks*, 10(5), 1999.
- [9] N. Cristianini, J. S. Taylor, "An introduction to support vector machines and other kernel-based learning methods", Cambridge UP, 2000.
- [10] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, Wiley and sons, 2001.
- [11] T. Joachims, SVMlight available at http://www.cs.cornell.edu/People/tj/svm_light
- [12] I. Laptev, T. Lindeberg, "Interest point detection and scale selection in space-time", to appear in *Proc. of Scale-Space'03*, Springer, LNCS, 2003.
- [13] B. Leibe, B. Schiele, "Analyzing appearance and contour based methods for object categorization", *CVPR'03*, 2003.
- [14] S. Z. Li, Q. Fu, L. Gu, B. Schölkopf, Y. Cheng, H. Zhang, "Kernel machine based learning for multi-view face detection and pose estimation", *Proc. ICCV'01*, 2001.
- [15] D. Lowe, "Object recognition from local scale invariant features", *Proc. ICCV'99*, 1999.
- [16] K. Mikolajczyk, C. Schmid, "An affine invariant interest point detector", *Proc. ECCV'02*, 2002.
- [17] S. A. Nene, S. K. Nayar and H. Murase, "Columbia Object Image Library (COIL-100)," TR CUCS-006-96, 1996.
- [18] M. Pontil, A. Verri, "Support vector machines for 3D object recognition", *IEEE TPAMI*, 20(6), 637-646, 1998.
- [19] D. Roobaert, M. Zillich, J. O. Eklundh, "A pure learning approach to background invariant object recognition using pedagogical support vector learning", *Proc. CVPR'01*, 2001.
- [20] F. Schaffalitzky, A. Zissermann "Viewpoint invariant texture matching and wide baseline stereo", *Proc. ICCV'01*, 2001.
- [21] B. Schiele, J. L. Crowley, "Recognition without correspondence using multidimensional receptive field histograms", *IJCV*, 36(1), 31-50, 2000.
- [22] C. Schmid, R. Mohr, "Local greyvalue invariants for image retrieval", *IEEE TPAMI*, 19(5), 530-535, 1997.
- [23] C. Schmid, R. Mohr, C. Bauckhage, "Evaluation of Interest Point Detectors", *IJCV*, 37(2), 151-172, 2000.
- [24] M. J. Swain, D. H. Ballard, "Color Indexing", *IJCV*, 7(1), 11-32, 1991.
- [25] V. Vapnik, *Statistical learning theory*, Wiley and sons, NY, 1998.
- [26] H. Bühlhoff, C. Wallraven, A. Graf, "View-based dynamic recognition based on human perception", *Proc. ICPR'02*, 2002.