Invariance in Kernel Methods by Haar-Integration Kernels

B. Haasdonk[1], A. Vossen[2], and H. Burkhardt[1]

[1] Computer Science Department,
Albert-Ludwigs-University Freiburg, 79110 Freiburg, Germany
{haasdonk, burkhardt}@informatik.uni-freiburg.de
[2] Institute of Physics, Albert-Ludwigs-University Freiburg,
79104 Freiburg, Germany
vossen@physik.uni-freiburg.de

Abstract. We address the problem of incorporating transformation invariance in kernels for pattern analysis with kernel methods. We introduce a new class of kernels by so called Haar-integration over transformations. This results in kernel functions, which are positive definite, have adjustable invariance, can capture simultaneously various continuous or discrete transformations and are applicable in various kernel methods. We demonstrate these properties on toy examples and experimentally investigate the real-world applicability on an image recognition task with support vector machines. For certain transformations remarkable complexity reduction is demonstrated. The kernels hereby achieve state-of-the-art results, while omitting drawbacks of existing methods.

1 Introduction

Many pattern analysis tasks are based on learning from examples, i.e. sets of observations are given, which are to be processed in some optimal way. Such tasks can consist of classification, regression, clustering, outlier-detection, feature-extraction etc. A powerful battery of algorithms for such tasks is given by so called *kernel-methods*, which attract increasing attention due to their generality, adaptability, theoretic foundation, geometric interpretability and excellent experimental performance, cf. [1]. The most famous representative is the support vector machine (SVM). It is meanwhile widely accepted, that additional problem specific prior knowledge is crucial for improving the generalization ability of such learning systems [2]. In particular, prior-knowledge about pattern transformations is often available. A simple example is that geometric transformations like rotations or translations of an image frequently do not change the inherent meaning of the displayed object. The insight, that the crucial ingredient for powerful analysis methods is the choice of a kernel function, led to various efforts of problem-specific design of kernel functions.

In this paper we introduce a new class of kernel-functions, so called Haar-integration kernels, which incorporate such transformation knowledge. They

H. Kalviainen et al. (Eds.): SCIA 2005, LNCS 3540, pp. 841–851, 2005.

are based on a successful technique for extracting invariant features, the so called Haar-integration procedure. Extension of this technique to kernel functions seems to be the first proposal, which omits various drawbacks of existing approaches. In particular the advantages are positive definiteness, steerable transformation extent, applicability in case of both continuous and discrete transformations, applicability to different kinds of base-kernel-functions and arbitrary kernel-methods.

The structure of the paper is as follows: In the next section we recall the required notions concerning kernel methods. Section 3 introduces the new proposal and derives some theoretical properties. We continue with comments on the relation to existing approaches. The subsequent Section 5 presents simple visualizations of the kernels in 2D. As sample kernel method we choose the SVM, for which we present some illustrative toy-classification results. In Section 6 real world applicability on an image recognition task is demonstrated consisting of a well known benchmark dataset for optical character recognition, the USPS digit dataset. Additionally, the kernels allow a remarkable speedup as is demonstrated in Section 7 before we finish with some concluding remarks.

2 Kernel Methods

In this section we introduce the required notions and notations, which are used in the sequel concerning kernel methods, cf. [1] for details on the notions and concepts. In general, a kernel method is a nonlinear data analysis method for patterns from some set $x \in \mathcal{X}$, which is obtained by application of the *kernel trick* on a given linear method: Assume some linear analysis method operating on vectors $\mathbf{x}$ from some Hilbert space $\mathcal{H}$, which only accesses patterns $\mathbf{x}$ in terms of inner products $\langle \mathbf{x}, \mathbf{x}' \rangle$. Examples of such methods are PCA, linear classifiers like the Perceptron, etc. If we assume some nonlinear mapping $\Phi : \mathcal{X} \to \mathcal{H}$, the linear method can be applied on the images $\Phi(x)$ as long as the inner products $\langle \Phi(x), \Phi(x') \rangle$ are available. This results in a nonlinear analysis method on the original space $\mathcal{X}$. The *kernel trick* now consists in replacing these inner products by a kernel function $k(x, x') := \langle \Phi(x), \Phi(x') \rangle$: As soon as the kernel function k is known, the Hilbert space $\mathcal{H}$ and the particular embedding Φ are no longer required. For suitable choices of kernel function k, one obtains methods, which are very expressive due to the nonlinearity, but cheap to compute, as explicit embeddings are omitted. If for some function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathcal{X} \to \mathcal{H}$ can be found such that $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$ holds, then k is called *positive definite (pd)*. A larger class of kernels which is useful for various kernel methods is the slightly weaker notion of *conditional* positive definite (cpd) kernels. Some standard kernels, which are used in practice for vectorial data $\mathbf{x}$ are the linear, polynomial, Gaussian and negative distance kernel, where the latter is cpd, the remaining ones are pd:

$$k^{\mathrm{lin}}(\mathbf{x}, \mathbf{x}') := \langle \mathbf{x}, \mathbf{x}' \rangle \qquad k^{\mathrm{nd}}(\mathbf{x}, \mathbf{x}') := -\left\| \mathbf{x} - \mathbf{x}' \right\|^{\beta}, \beta \in [0, 2] \tag{1}$$

$$k^{\mathrm{pol}}(\mathbf{x}, \mathbf{x}') := (1 + \gamma \langle \mathbf{x}, \mathbf{x}' \rangle)^{p} \qquad k^{\mathrm{rbf}}(\mathbf{x}, \mathbf{x}') := e^{-\gamma \left\| \mathbf{x} - \mathbf{x}' \right\|^2}, p \in \mathbb{N}, \gamma \in \mathbb{R}^{+}.$$

As a sample kernel method, we will refer to the SVM for classification. In the case of two-class classification, this method requires a kernel function k, training patterns and class labels $(x_i, y_i) \in \mathcal{X} \times \{\pm 1\}, i = 1, \ldots, n$ and produces a classification function which assigns the class $f(x) = \operatorname{sgn}\left(\sum_i \alpha_i y_i k(x, x_i) + b\right)$ to a new pattern x. Here the α_i, b are the parameters, which are determined during training. Multiclass problems can be solved by reducing a problem to a collection of two-class problems in various ways. We refrain from further details.

3 Haar-Integration Kernels

In the field of pattern recognition, particular interest is posed on invariant feature extraction, i.e. finding some function $I(x)$, which satisfies $I(x) \sim I(gx)$ or even with equality for certain transformations g of the original pattern x. One method for constructing such invariant features is the so called Haar-integration technique [3]. In this approach, invariant representations of patterns are generated by integration over the known transformation group. These features have been successfully applied on various real world applications ranging from images to volume data [4, 5, 6] and have been extended to be applicable on subsets of groups [7]. A similar technique can be applied to generate invariant kernels, which leads to the definition of the Haar-integration kernels.

Definition 1 (Haar-Integration Kernel). *Let $\mathcal{G}$ be a group operating on the set $\mathcal{X}$ with Haar-measure dg. Let $\mathcal{G}_0 \subset \mathcal{G}$ be a measurable subset. Let k_0 be a kernel on $\mathcal{X}$ such that for all x, x'*

$$k(x, x') = \int_{\mathcal{G}_0} \int_{\mathcal{G}_0} k_0(gx, g'x') dg dg' \tag{2}$$

exists and is finite. We denote this function the Haar-integration kernel (HI-kernel) *of k_0 with respect to $\mathcal{G}_0$.*

The requirement of the integrability of k_0 is practically mostly satisfied, e.g. after finite discretization of $\mathcal{G}_0$. The motivation of the integral (2) is demonstrated in Fig. 1 in two ways: a) in the original pattern space and b) in the k_0-induced feature space. For simplicity we assume the Haar-measure to be normalized to $dg(\mathcal{G}_0) = 1$. In the left figure, two patterns x, x' are illustrated in the pattern space $\mathcal{X}$ including their orbits $\mathcal{G}x, \mathcal{G}x'$. The goal is to find a kernel function, which satisfies $k(x, x') \sim k(gx, x')$ for small transformations g of x. If we define $\mathcal{G}_0$ as illustrated, the Haar-integration kernel generated by k_0 is the average over all pairwise combinations of $\mathcal{G}_0 x$ and $\mathcal{G}_0 x'$. If $\mathcal{G}_0$ is large enough, the integration ranges $\mathcal{G}_0 x$ and $\mathcal{G}_0 gx$ have a high overlap, which makes the resulting integrals arbitrarily similar. In the right sketch b), the interpretation of the kernels in feature-space is given: Instead of averaging over $k_0(x, x')$, the integration kernel is the inner product of the average of the sets $\Phi(\mathcal{G}_0 x')$, $\Phi(\mathcal{G}_0 x)$, respectively, due to

$$\left\langle \int_{\mathcal{G}_0} \Phi(gx) dg, \int_{\mathcal{G}_0} \Phi(g'x') dg' \right\rangle = \int_{\mathcal{G}_0} \int_{\mathcal{G}_0} \langle \Phi(gx), \Phi(g'x') \rangle \, dg dg' = k(x, x'). \tag{3}$$

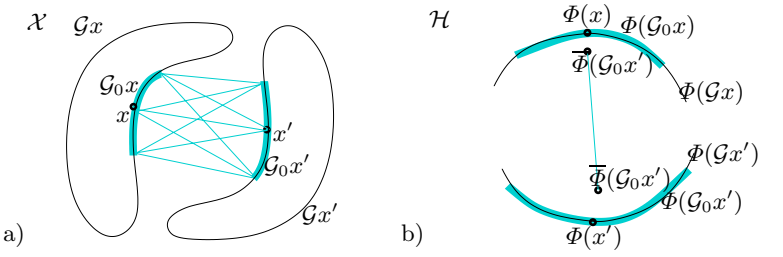

Fig. 1. Geometric interpretation of Haar-integration kernels. a) original pattern space $\mathcal{X}$, b) kernel-induced feature space $\Phi(\mathcal{X}) \subset \mathcal{H}$

Again, small transformations of x to gx results in similar sets of transformed patterns in feature space, similar averages and similar kernel values.

Some theoretical properties of these kernels are quite convenient:

Proposition 1 (Basic Properties of Haar-Integration Kernels).

(i) If $\mathcal{G}_0 = \mathcal{G}$ then k is invariant, i.e. $k(x, x') = k(gx, g'x')$ for all $x, x' \in \mathcal{X}, g, g' \in \mathcal{G}$.
(ii) If k_0 is a (c)pd kernel, then k is a (c)pd kernel.

Proof. (Sketch) (i) For characteristic functions $k_0(gx, g'x') = \chi_A(g) \cdot \chi_{A'}(g')$ with measurable $A, A' \subset \mathcal{G}$, we obtain with linearity of the integral and the invariance of the Haar-measure dg that $k(hx, h'x') = dg(A) \cdot dg'(A')$. This is independent of h, h', thus invariant. The invariance in case of these characteristic functions transfers similarly to other measurable sets $A \subset \mathcal{G} \times \mathcal{G}$, linear combinations of such characteristic functions and the limit operations involved in the Lebesgue-integral definition.

(ii) The symmetry of k is obvious. If k_0 is pd then $\bar{\Phi}(x) := \int_{\mathcal{G}0} \Phi(gx) dg$ is a mapping from $\mathcal{X}$ to $\mathcal{H}$ with $\left\langle \bar{\Phi}(x), \bar{\Phi}(x') \right\rangle = k(x, x')$ according to (3). So in particular k is pd. If k_0 is cpd., the kernel $\tilde{k}_0$ following [1–Prop. 2.22] is pd (and cpd), so is the corresponding HI-kernel $\tilde{k}$. This function contains the HI-kernel k of k_0 plus some functions depending on solely one of the arguments x, x'. Such functions maintain cpd-ness, so k is cpd. □

The HI-kernels are conceptionally an elegant seamless connection between non-invariant and invariant data-analysis: The size of $\mathcal{G}_0$ can be adjusted from the non-invariant case $\mathcal{G}_0 = \{\mathrm{id}\}$, which recovers the base-kernel, to the fully invariant case $\mathcal{G}_0 = \mathcal{G}$. This will be further demonstrated in Sec. 5.

4 Relation to Existing Approaches

We want to discuss some relations to existing feature-extraction and invariant SVM methods. Eqn. (3) indicates the relation of the HI-kernels to the *partial Haar-integration features* [7]. The HI-kernels are inner products of corresponding

Haar-integral features in the Hilbert space $\mathcal{H}$. Some kernels are known to induce very high or even infinite dimensional spaces, e.g. k^{rbf}. So in these cases, the HI-kernels represent Haar-integral invariants of infinite dimension. Clearly this is a computational advantage, as these could not be computed explicitly in feature space. On the other hand, all inner products between Haar-invariant feature representations are captured by the HI-kernel approach, by a suitable base-kernel k_0. So these kernels are conceptionally richer.

Invariance in kernel methods has been mainly proposed resulting in non-positive definite kernels as the *jittering kernels* [2], *tangent distance kernels* [8] or *tangent vector kernels* [9]. In contrast to these methods, the proposed kernels have the important advantage of being positive definite. They can be applied to non-differentiable, discrete transformations and to general kernels, not only to distance-based ones or the Gaussian. In contrast to [10], which theoretically constructs invariant kernels by solving partial differential equations, we obtain practically applicable kernels.

There are further methods of specially incorporating invariance in SVM. The method of *invariant hyperplane* or the nonlinear extension *invariant SVM* [11, 12] are theoretical nice constructions of enforcing the invariant directions into the SVM optimization problem. However they suffer from the high computational complexity. The most widely accepted method for invariances in SVM is the *virtual support vector (VSV)* method [13]. It consists of a two step training stage. In a first ordinary SVM training step, the support vectors are determined. This set is multiply enlarged by various small transformations of each support vector. The second training stage on this set of virtual support vectors yields an invariant SVM. The problem of this approach is the enlarged memory and time complexity during training.

5 Toy-Experiments

In this section we illustrate the kernels and their application in a kernel-method on simple toy-examples. For this, we choose the patterns $\mathbf{x}$ from the Euclidean plane $\mathcal{X} = I\!R^2$, and define simple transformations on these points. The transformations are translations along fixed directions, rotation around a fixed point, shift along a sinus-shaped curve and reflection along a vertical axis. By these transformations we cover linear, nonlinear and extremely nonlinear operations. Additionally, they represent both continuous and discrete transformations.

We start with illustration of the kernel functions in Fig. 2. For this, we fix one point $\mathbf{x}'$ (black dot) and plot the kernel value $k(\mathbf{x}, \mathbf{x}')$, while $\mathbf{x}$ varies over the unit-square. The kernel values are color-coded. We start with the demonstration of the invariant linear and polynomial kernel the upper row. Subplot a) demonstrates the non-invariant kernel k^{lin}, which is made invariant with respect to reflections along a perpendicular axis in b). Subplot c) illustrates the kernel k^{pol} (of degree 2), which is nicely made invariant with respect to complete rotations d). Here and in the subsequent plots, we highlight the integration range $\mathcal{G}_0\mathbf{x}'$ by solid lines passing through the point $\mathbf{x}'$. So these inner-product

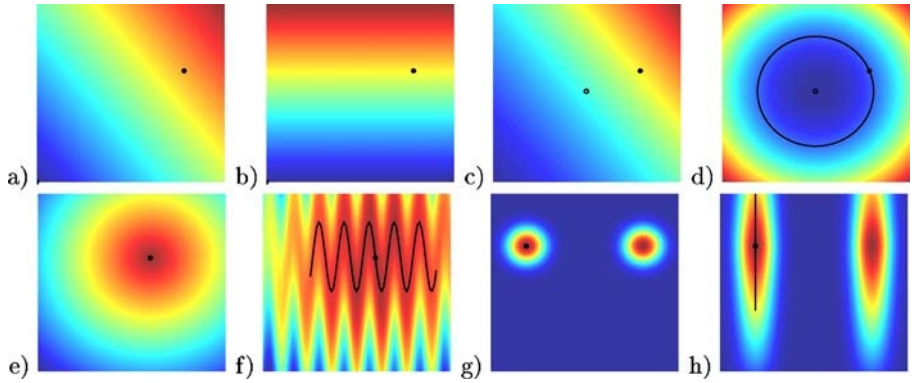

Fig. 2. Demonstration of invariant kernels. a), b) non-invariant/reflection invariant k^{lin}, c), d) non-invariant/rotational invariant k^{pol}, e), f) k^{nd} with highly nonlinear sinus invariance, g), h) k^{rbf} with simultaneous reflection and translation invariance

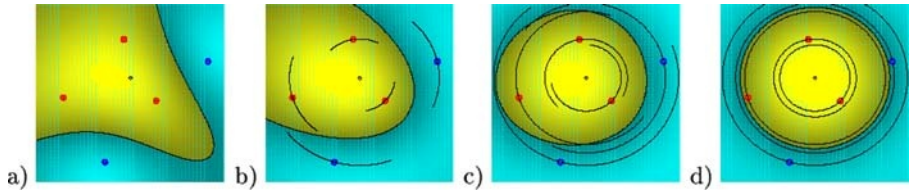

Fig. 3. Demonstration of invariant kernels in SVM classification. a) non-invariant k^{rbf}, b), c) partial rotational invariance, d) complete rotational invariance

kernels work nicely for these global transformation groups. However it turned out, that they have problems with partial invariances, e.g. reducing the range of rotations in d). The reason is, that the required nonlinearity increases: Reducing the circle of rotated patterns to a semi-circle, the isolines of the desired invariant kernel would need to be half-moon-shaped around the semi-circle. This cannot be expressed by a HI-kernel of a 2nd degree polynomial, as averaging does not increase the polynomial degree. So ideally, base-kernels are required, which can express arbitrary complex boundaries. Such kernels are given by k^{nd} or k^{rbf}. These kernels proved to work in all preceding cases and cases which we present in the lower row. Plot e) and f) illustrate the negative distance kernel ($\beta = 1$) for highly nonlinear transformations consisting of shifts along sinus-curves, where the size of $\mathcal{G}_0$ can be smoothly increased between the plots. Similarly, the Gaussian kernel is made invariant with respect to the combination of reflection and increasing y-translations in g) and h). In both cases the transformations are nicely captured covering linear, highly nonlinear, discrete transformations and combinations thereof.

We continue with demonstrating the increased separability when applied in classification problems by SVM. Given 5 (red and blue) points in Fig. 3, the effect of increasing rotational invariance (with respect to the black circle) is illustrated.

In the leftmost non-invariant case, the classification result of a standard k^{rbf} is illustrated, which correctly classifies the points, but indeed captures none of the rotational invariance. By increasing the rotational integration range, the solution is a completely invariant SVM solution in d). Similar results can be obtained for the negative distance kernel.

6 Real-World-Experiments

As a real world application of the kernels, we perform image classification experiments on an optical character recognition problem. In this setting partial invariances are particularly useful as e.g. only small rotations are allowed, whereas large rotations will confuse W and M, 6 and 9 or N and Z. Only small x- and y-translations are reasonable, if the patterns are already roughly centered. We restrict the real world experiments to these rigid transformations and the k^{rbf} kernel, as this turned out to capture nicely partial invariances in the previous toy-experiments.

For enabling comparisons with existing methods, we chose the well known benchmark dataset of USPS-digits. The corpus consists of 7291 training and 2007 test images of 16×16 greyvalue images of handwritten digits. Figure 4 depicts a) some simple and b) difficult to classify example images. A list of reference results can be found in [1]. The relevant results for our purpose are the 2.5% test error rate, which is reported for humans [14] and indicates the difficulty of the dataset. A standard polynomial SVM is reported to reach 4.0% error rate [15], which is improved by the VSV-method involving translations obtaining 3.2% test error [13]. This is the most comparable result in literature. Some better results have been reported, but those apply more sophisticated deformation models, more training data etc. So they involve different kinds of prior knowledge than only rigid transformations of the images.

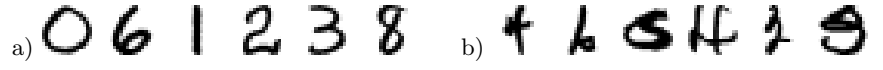

Fig. 4. Examples of USPS digits. a) easy, b) difficult to classify examples

As classifier for the dataset, we use a one-versus-rest multiclass SVM applying the HI-kernel (HI-SVM). The SVM package LIBSVMTL [16] was taken as a basis for the implementation. In the first set of experiments we focus on recognition accuracy. After setting the integration ranges, the remaining SVM-parameter pair is (C, γ). For this we chose 10^2 combinations of 10 values for each parameter. One SVM model was trained for each parameter set and the results of the best models are reported in the following. Note that this is a kind of *model selection on the test set* which produces optimistically biased test-error rates compared to the true generalization performance. But this is a general phenomenon for many public datasets including USPS.

Table 1. USPS recognition results with HI-kernels. rotation integration (left), x-y-translation (right)

ϕ-range [rad]	k^{rbf} test error [%]
0	4.5
$\pm 0.04\pi$	4.1
$\pm 0.08\pi$	4.2
$\pm 0.12\pi$	3.9
$\pm 0.16\pi$	4.2

x-y-range [pixels]	k^{rbf} test error [%]
0	4.5
± 1	3.7
± 2	3.2
± 3	3.3
± 4	3.2

The left part of Tab. 1 lists the results which are obtained by increasing the rotation integration. Numerical integration is performed involving 3×3 sample points for each $\mathcal{G}_0$ integral. The table indicates that the HI-kernel clearly capture the wanted invariances, as the results improve compared to the non-invariant SVM, which has 4.5% test error. Still the results are far from the existing VSV-result, which was based on translations. Therefore, a second experiment sequence was performed by regarding translations only. Initial experiments yielded that with increasing the number of integration evaluation points to 9×9 very good results are obtained. We increase the translation range from 0 to ± 4 pixels. The results of this are depicted in the left part of Tab. 1. The results clearly improve and equal the state-of-the art result of the VSV approach.

7 Acceleration

The integration kernels allow various ways of time complexity reduction. Suitable caching strategies can accelerate the computation procedure, e.g. caching the transformed patterns throughout the computation of the kernel-matrix, etc. A more fundamental complexity reduction can be performed similar as in the case of jittering kernels, if the transformations are commutative and compatible with the base-kernel in the sense that $k_0(gx, g'x') = k_0(x, g^{-1}g'x)$. An example of such kernels are all kernels based on distance or inner products of vectors, if the transformations are rotations, translations or even permutations of the vector entries. In this case, the complexity of a single kernel evaluation can be reduced remarkably from $\mathcal{O}(s^{2l})$ to $\mathcal{O}(s^l)$, where s is the number of integration steps along each of the l transformation directions. This integral reduction can be obtained by halving the number of integrals:

$$\int_{\mathcal{G}_0}\int_{\mathcal{G}_0} k_0(gx, g'x')dgdg' = \int_{\mathcal{G}_0}\int_{\mathcal{G}_0} k_0(x, g^{-1}g'x')dgdg' = \int_{\mathcal{G}_0^{-1}\mathcal{G}_0} k_0(x, \bar{g}x')d\bar{g}. \quad (4)$$

where $\mathcal{G}_0^{-1}$ denotes the set of all inverted $\mathcal{G}_0$ elements and $d\bar{g}$ denotes a suitable resulting measure. If $\mathcal{G}_0$ is chosen reasonable, the resulting set $\mathcal{G}_0^{-1}\mathcal{G}_0$ will be much smaller than the original $\mathcal{G}_0 \times \mathcal{G}_0$. We continue with a second method of

Table 2. Complexity comparison between HI-SVM and VSV-method with 3×3 2D-translations

Method	test-error [%]	train-time [s]	test-time [s]	average #SV
HI-SVM	3.6	1771	479	412
HI-SVM, IR	3.6	810	176	412
HI-SVM, SV	3.6	113 + 130 + 297	466	410
HI-SVM, SV + IR	3.6	113 + 130 + 91	172	410
VSV-SVM	3.5	113 + 864 + 1925	177	4240

acceleration in the special case of SVM-classification. The support-vectors of the non-invariant SVM and the HI-SVM turn out to have a high overlap. This suggests to apply the idea of the VSV-SVM on HI-SVM: Perform a two-step training stage by initial ordinary k^{rbf} SVM training, then selecting the support vectors and performing an HI-SVM training on this SV-set.

We performed tests of all combinations of these two acceleration methods denoted as IR (integral reduction) and SV (support vector extraction) in Tab. 2, and investigated the resulting time and model complexities. For comparison, we added the VSV-model complexities. In order not to bias the results towards one of the methods, we fixed $C = 100, \gamma = 0.01$ and the x-y-translation to ± 2 pixels, with (implicit) 3×3 transformed patterns per sample. The recognition results are almost identical, but not very expressive as they are suboptimal due to the missing C, γ optimization. The experiments indicate, that the integral reduction (IR) indeed reduces the training and test-time remarkably, while the recognition accuracy remains unchanged as expected. By applying the SV-extraction step in an initial training stage, the error rate does not increase, but the training time (first training + (V)SV-extraction + second training) is again largely reduced. The testing time does only improve marginally by SV-extraction as the (rounded) number of SV in the final models is not significantly reduced. The comparison between the accelerated Haar-integral methods and the VSV yields, that the model size in terms of average number of SVs is clearly smaller applying our kernels, as the kernels themselves represent relevant information of the model by the invariance, so storage complexity is largely reduced. The HI-kernels are more expensive to evaluate than standard kernels as the VSV method uses, so testing is more expensive than the VSV-method. Still, by the acceleration methods the testing time can compete with the VSV-method. In the training time, the high value of 864 sec for the SV-extraction is to be taken with care and might be decreased by optimizing the explicit construction of the VSV set. Despite this possible optimization, the accelerated integration kernels are during training still clearly faster than the VSV-SVM. This is due to the fact, that the VSV method suffers from the 9-times enlarged SV-training set. Although both the VSV and the fast HI-kernels are expected to slowdown quadratically, the VSV seems to be more affected by this.

8 Conclusions

We introduced a class of invariant kernels called Haar-integration kernels. These kernels seem to be the first invariant kernels, which alleviate the problem of missing positive definiteness, as observed in other approaches. Furthermore, they are not restricted to continuous or differentiable transformations but allow explicit discrete or continuous transformations. The degree of invariance can be smoothly adjusted by the size of the integration interval. Experimental application in a particular kernel method, namely SVM, allowed a comparison to the state-of-the art method of VSV. Test on the real world USPS dataset demonstrates that state-of-the art recognition results can be obtained with these kernels. Complexity comparisons demonstrated large improvements in training and testing time by the techniques of integral reduction and training on the SVs of an ordinary k^{rbf}-SVM. The expensive HI-kernel evaluation is ameliorated by the reduced model size during testing, such that both training and testing times can compete with or outperform the VSV approach.

So in SVM learning the kernels seem a good alternative to VSV, if the testing time is not too crucial or small model size is required. In other kernel methods where the complexity grows quadratically with the number of training examples, the generation and storing of virtual examples might be prohibitive. In these situations, the proposed kernels can be a welcome approach. Perspectives are to apply the integration kernels in further kernel-methods and on further datasets. Interesting options are, to apply the technique to non-group transformations, in particular non-reversible transformations as long as "forward" integrations are possible.

References

1. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press (2002)
2. DeCoste, D., Schölkopf, B.: Training invariant support vector machines. Machine Learning **46** (2002) 161–190
3. Schulz-Mirbach, H.: Constructing invariant features by averaging techniques. In: Proc. of the 12th ICPR. Volume 2., IEEE Computer Society (1994) 387–390
4. Schael, M.: Texture defect detection using invariant textural features. In: Proc. of the 23rd DAGM - Symposium Mustererkennung, Springer Verlag (2001) 17–24
5. Siggelkow, S.: Feature-Histograms for Content-Based Image Retrieval. PhD thesis, Albert-Ludwigs-Universität Freiburg (2002)
6. Ronneberger, O., Schultz, E., Burkhardt, H.: Automated pollen recognition using 3d volume images from fluorescence microscopy. Aerobiologia **18** (2002) 107–115
7. Haasdonk, B., Halawani, A., Burkhardt, H.: Adjustable invariant features by partial Haar-integration. In: Proc. of the 17th ICPR. Volume 2. (2004) 769–774
8. Haasdonk, B., Keysers, D.: Tangent distance kernels for support vector machines. In: Proc. of the 16th ICPR. Volume 2. (2002) 864–868
9. Pozdnoukhov, A., Bengio, S.: Tangent vector kernels for invariant image classification with SVMs. In: Proc. of the 17th ICPR. (2004)
10. Burges, C.J.C.: Geometry and invariance in kernel based methods. In: Advances in Kernel Methods - Support Vector Learning, MIT Press (1999) 89–116

11. Chapelle, O., Schölkopf, B.: Incorporating invariances in nonlinear support vector machines. In: NIPS 14, MIT-Press (2002) 609–616
12. Schölkopf, B., Simard, P., Smola, A., Vapnik, V.: Prior knowledge in support vector kernels. In: NIPS 10, MIT Press (1998) 640–646
13. Schölkopf, B., Burges, C., Vapnik, V.: Incorporating invariances in support vector learning machines. In: ICANN'96, LNCS, 1112, Springer (1996) 47–52
14. Simard, P., LeCun, Y., Denker, J.: Efficient pattern recognition using a new transformation distance. In: NIPS 5, Morgan Kaufmann (1993) 50–58
15. Schölkopf, B., Burges, C., Vapnik, V.: Extracting support data for a given task. In: Proc. of the 1st KDD, AAAI Press (1995) 252–257
16. Ronneberger, O., Pigorsch, F.: LIBSVMTL: a support vector machine template library (2004) http://lmb.informatik.uni-freiburg.de/lmbsoft/libsvmtl/.