Building Kernels From Binary Strings for Image Matching

Francesca Odone, Annalisa Barla, and Alessandro Verri

***Abstract*—In the statistical learning framework, the use of appropriate kernels may be the key for substantial improvement in solving a given problem. In essence, a kernel is a similarity measure between input points satisfying some mathematical requirements and possibly capturing the domain knowledge. In this paper, we focus on kernels for images: we represent the image information content with binary strings and discuss various bitwise manipulations obtained using logical operators and convolution with nonbinary stencils. In the theoretical contribution of our work, we show that histogram intersection is a Mercer's kernel and we determine the modifications under which a similarity measure based on the notion of Hausdorff distance is also a Mercer's kernel. In both cases, we determine explicitly the mapping from input to feature space. The presented experimental results support the relevance of our analysis for developing effective trainable systems.**

I. Introduction

A MAIN issue of statistical learning approaches, like support vector machines or other kernel methods [1], [2] is which kernel to use for which problem. A number of general-purpose kernels are available in the literature, but, especially in the case in which only a relatively small number of examples is available, there is little doubt that the use of an appropriate kernel function can lead to a substantial increase in the generalization ability of the developed learning system. Ultimately, the choice of a specific kernel function is based on the prior knowledge of the problem domain.

In this paper, we discuss kernels based on the manipulation of binary vectors in the computer vision domain, considering both histogram and iconic representation. Since binary vectors can also be used for representing the information content of signals other than images, the presented analysis may be of interest to a number of application domains.

The difficulty of building kernels is to satisfy certain mathematical requirements while incorporating the prior knowledge of the application domain. For this reason, we studied different similarity measures for image matching and determined under what assumptions they can be used to build kernels for images. Since the image kernels we found are based on manipulating binary strings, we first present the results of our work discussing kernels for binary strings in abstract terms and then we specialize our results to the computer vision domain.

The topic of kernels for images is relatively new. A number of studies have been reported about the use of general purpose kernels for image-based classification problems [3]–[8]. One of the first attempts to introduce prior knowledge on the correlation properties of images in the design of an image kernel can be found in [9]; in [10], instead, a family of functions which seem to be better suited than Gaussian kernels for dealing with image histograms has been studied. In essence, the kernels described in [10] ensure heavier tails than Gaussian kernels in the attempt to contrast the well-known phenomenon of diagonally dominant kernel matrices in the case of high-dimensional inputs. In [11], an RBF kernel is engineered in order to use the Earth Mover's Distance as a dissimilarity measure, while in [12], an RBF kernel based on χ^2 has been shown to be of Mercer's type. As opposed to the global approach to image representation which is common to all of the previous works, in [13], a class of Mercer's kernels for local image features is proposed.

This paper is organized as follows. In Section II, we set the background and describe the mathematical requirements for kernel functions. In Section III, we discuss the various manipulations on binary strings which we studied. Kernels for histogram-based representations are discussed in Section IV, while kernels for iconic representations are the theme of Section V. Experimental results, including extensive comparison with previous work, are reported in Section VI. We draw the conclusions of our work in Section VII.

II. Kernel Functions

In this section, we summarize the mathematical requirements a function needs to satisfy in order to be a legitimate kernel for SVMs.

A. Support Vector Machines

Following [14], many problems of statistical learning [15], [1] can be cast in an optimization framework in which the goal is to determine a function minimizing a functional I of the form

$$I[f] = \frac{1}{\ell}\sum_{i=1}^{\ell} V(f(\mathbf{x}_i), y_i) + \lambda \|f\|_K^2 \qquad (1)$$

where the ℓ pairs $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_\ell, y_\ell)\}$, the examples, are i.i.d. random variables drawn from the space $X \times Y$ according to some fixed but unknown probability distribution, V is a *loss* function measuring the fit of the function f to the

Manuscript received October 17, 2003; revised March 23, 2004. This work was supported in part by the Istituto Nazionale di Fisica della Materia Advanced Research Project MAIA, in part by the FIRB Project ASTA RBAU01877P, and in part by the EU Project IST-2000-25341 KerMIT. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Christophe Molina.

The authors are with the Istituto Nazionale di Fisica della Materia and with DISI, Universita' di Genova, I-16146 Genova, Italy (e-mail: odone@disi.unige.it; barla@disi.unige.it; verri@disi.unige.it).

Digital Object Identifier 10.1109/TIP.2004.840701

data, $\|\cdot\|_K$ the norm of f induced by a certain function K, named *kernel*, controlling the smoothness—or capacity—of f, and $\lambda > 0$ a tradeoff parameter. For several choices of the loss function V, the minimizer of the functional in (1) takes the general form

$$\sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \tag{2}$$

where the coefficients α_i depend on the examples. The mathematical requirements on K must ensure the convexity of (1) and hence the uniqueness of the minimizer (2).

SVMs for classification [15], for example, correspond to choices of V like

$$V(f(\mathbf{x}_i), y_i) = |1 - y_i f(\mathbf{x}_i)|_+$$

with $|t|_+ = t$ if $t \geq 0$, and 0 otherwise, and lead to a convex QP problem with linear constraints in which many of the α_i vanish. The points $\mathbf{x}_i$ for which $\alpha_i \neq 0$ are termed *support vectors* and are the only examples needed to determine the solution (2).

Before discussing the mathematical requirements on K, we briefly consider two special cases of classifiers that we will use for our experiments: binary classifiers and novelty detectors, or one-class classifiers.

B. Binary Classification

In the case of binary classification [15], we have $y_i \in \{-1, 1\}$ for $i = 1, \ldots, \ell$, and the dual optimization problem can be written as

$$\begin{aligned} \max_{\alpha_i} \quad & \sum_{i=1}^{\ell} |\alpha_i| - \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C \text{ if } y_i = 1 \\ & -C \leq \alpha_i \leq 0 \text{ if } y_i = -1. \end{aligned} \tag{3}$$

A new point is classified according to the sign of the expression

$$\sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b$$

where the coefficient b can be determined from the Kuhn Tuker conditions. A closer look to the QP problem (3) reveals that the uniqueness of the solution is ensured by the convexity of the objective function.

C. Novelty Detection

In the case of novelty detection described in [16], for all training points, we have $y_i = 1$—i.e., all the training examples describe only one class—and the optimization problem reduces to

$$\begin{aligned} \max_{\alpha_i} \quad & \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \alpha_i = 1 \\ & 0 \leq \alpha_i \leq C. \end{aligned} \tag{4}$$

If $K(\mathbf{x}, \mathbf{x})$ is constant over the domain X, a novelty is detected if the inequality

$$\sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \geq \tau \tag{5}$$

is violated for some fixed value of the threshold parameter $\tau > 0$. Similarly to the case of above, the uniqueness of the solution is ensured by requiring the convexity of the quadratic form in the objective function.

D. Kernel Requirements

The convexity of the functionals (3) and (4) is guaranteed if the function K is positive definite. Remember that a function $K : X \times X \to \mathbb{R}$ is *positive definite* if for all n and all choices of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in X$, and $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{R}$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0. \tag{6}$$

A theorem of functional analysis due to Mercer [17] allows us to write a positive definite function as an expansion of certain functions $\phi_k(\mathbf{x}) k = 1, \ldots, N$, with N possibly infinite, or

$$K(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{N} \phi_k(\mathbf{x}) \phi_k(\mathbf{x}'). \tag{7}$$

For this reason, a positive definite function is also called a *Mercer's kernel*, or simply a *kernel*. Whether or not this expansion is explicitly computed, for any kernel, $K, K(\mathbf{x}, \mathbf{x}')$ can be thought of as the inner product between the vectors $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_N(\mathbf{x}))$, and $\phi(\mathbf{x}') = (\phi_1(\mathbf{x}'), \phi_2(\mathbf{x}'), \ldots, \phi_N(\mathbf{x}'))$. For a rigorous statement of Mercer's theorem, the interested reader is referred to [17], [18].

III. Manipulating Binary Strings

In this section, we discuss the manipulations of binary strings that we studied and applied to images. We start by establishing the notation we use throughout the paper.

A. Notation

We denote strings with upper-case letters like A and B and bits with lower-case letters like a and b. A binary string of fixed length is a sequence of bits like 0010 or $a_1 a_2 \ldots a_P$ with $a_p \in \{0, 1\}$ for $p = 1, \ldots, P$. A string A of length P can also be written as a P-dimensional vector $\mathbf{A} = (a_1, \ldots, a_P)$. If $\mathbf{A}$ is an ordinary P-dimensional vector, the components of which can take on arbitrary values, we write $\mathbf{A} = (A_1, \ldots, A_P)$. Without risk of confusion, we do not distinguish between the *truth* values "0" and "1" and the *numerical* values "0" and "1." Finally, we

write $\mathbf{A} \cdot \mathbf{B}$ for the standard inner product between the two vectors $\mathbf{A}$ and $\mathbf{B}$.

B. Making Kernels With Logical Connectives

We first deal with the case in which all binary strings have the same length and the similarity between strings is computed bitwise. As we will see in the next section, this analysis is relevant to histogram-like representations. The question we address is which two-place logical operator acting bitwise on pairs of binary strings A and B defines a Mercer's kernel. We start by counting the number of operators which can be defined on bit pairs. Since there are four possible input bit pairs—$(0,0), (0,1), (1,0)$, and $(1,1)$—and two possible output values—0 and 1—simple counting shows that there are only 2^4 different logical operators on bit pairs. Following standard notation, we use $\wedge, \vee$, and $\leftrightarrow$ for the *conjunction*, *disjunction*, *if-and-only-if* two-place logical operators, and $\neg$ for the *negate* one-place operator. We start with the conjunction, the AND operator, and define

$$K_{\wedge}(A,B) = \sum_{p=1}^{P} a_p \wedge b_p. \tag{8}$$

It is easy to see that $K_{\wedge}$ is the *linear kernel* acting on the vectors $\mathbf{A}$ and $\mathbf{B}$ since

$$\sum_{p=1}^{P} a_p \wedge b_p = \sum_{p=1}^{P} a_p b_p = \mathbf{A} \cdot \mathbf{B}.$$

Of the remaining 15 cases, only the *negate of the disjunction*, the NOR operator, and the *if-and-only-if* operator define Mercer's kernels. Indeed, using the identity

$$\neg(a_p \vee b_p) \equiv (\neg a_p) \wedge (\neg b_p)$$

we find that

$$\begin{aligned} K_{\neg\vee}(A,B) &= \sum_{p=1}^{P} \neg(a_p \vee b_p) = \sum_{p=1}^{P} (1-a_p)(1-b_p) \\ &= (\neg\mathbf{A}) \cdot (\neg\mathbf{B}) \end{aligned}$$

from which we see that $K_{\neg\vee}$ is the standard inner product between the binary vectors obtained by negating each component of the original vectors $\mathbf{A}$ and $\mathbf{B}$, respectively. Furthermore, from the fact that the sum of kernels is a kernel, we find that

$$K_{\leftrightarrow}(A,B) = \sum_{p=1}^{P} a_p \leftrightarrow b_p$$

is a Mercer's kernel since

$$K_{\leftrightarrow}(A,B) = K_{\wedge}(A,B) + K_{\neg\vee}(A,B).$$

It can be immediately verified through straightforward counterexamples that the other 13 two-place logical operators do not define Mercer's kernels. Let us consider, for example, $\otimes$, the *or exclusive* operator, and define

$$K_{\otimes}(A,B) = \sum_{p=1}^{P} a_p \otimes b_p = \mathbf{A} \otimes \mathbf{B}.$$

Explicit computation of the 2×2 matrix K, $K_{ij} = \mathbf{A}_i \otimes \mathbf{A}_j$ for the two binary strings $\mathbf{A}_1 = (1,0)$ and $\mathbf{A}_2 = (0,1)$ gives

$$K = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$$

the determinant of which is negative. We, thus, have that $K_{\otimes}$ is not a Mercer's kernel.

C. Beyond Bitwise Matching

Up to now, we have discussed operators acting bitwise on binary strings. If nearby bits in a string are correlated, we aim at studying Mercer's kernels able to capture short range correlations. These kernels are expected to be particularly useful when dealing with images and signals in the original domain.

The key idea is to replace the binary vector $\mathbf{A}$ with the nonbinary vector $\mathbf{A}^S$. The vector $\mathbf{A}^S$ can be thought of as the result of convolving $\mathbf{A}$ with the stencil $\mathbf{S}$[1]

$$\mathbf{A}^S = \mathbf{A} * \mathbf{S}. \tag{9}$$

Intuitively, the vector $\mathbf{A}^S$ captures short range spatial correlations between nearby components of $\mathbf{A}$. For example, if $\mathbf{S} = (1,1,1)$, we have

$$A_p^S = \sum_{i=-1}^{1} a_{p+i}$$

where, for simplicity, we consider the last component a_P adjacent to the first a_1 (equivalently, all summations on subscripts are taken modulo P). We immediately see that each stencil $\mathbf{S}$ identifies a Mercer's kernel K_S, or

$$K_S(A,B) = \mathbf{A}^S \cdot \mathbf{B}^S$$

where the convolution in (9) defines the feature mapping.

We now show that a stencil satisfying some additional constraints leads to a new type of Mercer's kernel which, as we will see in Section V, is well suited for iconic matching.

Theorem: Let $\mathbf{S}$ be defined as

$$\mathbf{S} = (\underbrace{1/(2m), \ldots, 1/(2m)}_{m}, 1, \underbrace{1/(2m), \ldots, 1/(2m)}_{m}) \tag{10}$$

then there exists a mapping $\phi : \mathbb{R}^P \rightarrow \mathbb{R}^{2m \times P}$ defined as $\tilde{\mathbf{V}} = \phi(\mathbf{V})$ such that for all vectors $\mathbf{V} \in \mathbb{R}^P$

$$\mathbf{V}^S \cdot \mathbf{V} = \tilde{\mathbf{V}} \cdot \tilde{\mathbf{V}}.$$

[1]A *stencil* is a pattern of weights used in computing a convolution, arranged in such a way as to suggest the spatial relationships between the places where the weights are applied [19].

Proof: Using the stencil $\mathbf{S}$ of (10) for the components $p = 1, \ldots, P$ of $\mathbf{V}^S$, we find

$$V_p^S = v_p + \frac{1}{2m} \sum_{i=-m_{i\neq 0}}^{m} v_{p+i}.$$

Taking the inner product between $\mathbf{V}^S$ and $\mathbf{V}$ gives

$$\mathbf{V}^S \cdot \mathbf{V} = \frac{1}{4m} \left(4m \sum_{p=1}^{P} v_p^2 + 2 \sum_{p=1}^{P} \sum_{i=-m_{i\neq 0}}^{m} v_p v_{p+i} \right).$$

Expanding and rearranging the terms of the above sums, we obtain

$$\begin{aligned} 4m\mathbf{V}^S \cdot \mathbf{V} = {} & \underbrace{v_1^2 + \cdots + v_1^2}_{4m} + \cdots + \underbrace{v_P^2 + \cdots + v_P^2}_{4m} \\ & + 2v_1 \underbrace{(v_{1-m} + \cdots + v_{1+m})}_{2m} + \cdots \\ & + 2v_P \underbrace{(v_{P-m} + \cdots + v_{P+m})}_{2m}. \end{aligned}$$

Finally, grouping the squares and the corresponding mix products gives

$$\begin{aligned} \mathbf{V}^S \cdot \mathbf{V} = \frac{1}{4m} \Bigg(& \sum_{i=-m_{i\neq 0}}^{m} (v_1 + v_{1+i})^2 + \cdots \\ & + \sum_{i=-m_{i\neq 0}}^{m} (v_P + v_{P+i})^2 \Bigg). \end{aligned} \quad (11)$$

From (11), it follows that the dot product $\mathbf{V}^S \cdot \mathbf{V}$ is the sum of $2m \times P$ squares and can, thus, be regarded as the square of the norm of a vector $\tilde{\mathbf{V}}$. ■

The $2m \times P$ components of $\tilde{\mathbf{V}}$, which are of the type

$$\frac{1}{2\sqrt{m}}(v_p + v_{p+i})$$

for $p = 1, \ldots, P$ and $i = -m, \ldots, m$ $(i \neq 0)$, can be thought of as features[2] of a feature mapping underlying the Mercer's kernel H_S defined as

$$H_S(\mathbf{A}, \mathbf{B}) = \tilde{\mathbf{A}} \cdot \tilde{\mathbf{B}}.$$

Simple counting arguments can be advocated to extend this theorem to other stencils subject to the condition that the central entry is 1 and the remaining $\mathcal{M}$ entries w_i can be written as $w_i = n_i/\mathcal{N}$ with $\mathcal{N} = \sum_{j=1}^{\mathcal{M}} n_j$.

We conclude by observing that, somewhat surprisingly, the kernel H_S can also be written as

$$H_S(\mathbf{A}, \mathbf{B}) = \frac{1}{2}(\mathbf{A} \cdot \mathbf{B}^S + \mathbf{B} \cdot \mathbf{A}^S). \quad (12)$$

[2]For the specific stencil $\mathbf{S}$ in the theorem, every component appears twice on the r.h.s. of (11) and the mapping could be more parsimoniously defined in a feature space of dimensionality $m \times P$. In general, however, the dimensionality of the feature space is $2m \times P$.

As we will see in Section V, this writing is closely related with a similarity measure proposed for iconic matching in [20].

We are now in a position to study under what conditions a number of similarity measures, well known to the computer vision community, can be used as kernels in statistical learning methods. We start discussing the case of images represented through histograms.

IV. Histogram Intersection Kernel

In this section, we focus on histogram intersection, a technique made popular in [21] for color indexing with application to object recognition.

We start off by reminding the definitions of histograms for images and histogram intersection: in the case of images a *histogram* can be associated to properties or characteristics of each pixel, such as the brightness or the color of pixels, the direction or the magnitude of edges, etc. After defining the range of possible values for a given property and dividing the range in intervals (called *bins*), the histogram is computed by counting how many pixels of the image belong to each interval.

Histogram intersection can be used as a similarity measure for histogram-based representations of images. In the simple case of one-dimensional (1-D) images of N pixels, we denote with A and B the M-bin histograms of images $\mathcal{A}$ and $\mathcal{B}$, respectively, and define the histogram intersection H_{int} as

$$H_{\text{int}}(A, B) = \sum_{m=1}^{M} \min\{A_m, B_m\} \quad (13)$$

with A_m and B_m the mth bin for $m = 1, \ldots, M$ and $\sum_{m=1}^{M} A_m = \sum_{m=1}^{M} B_m = N$. Intuitively, histogram intersection measures the degree of similarity between two histograms and we clearly have $0 \leq H_{\text{int}} \leq N$. Unlike other similarity measures, like L^2 distance, for example, this similarity measure is not only well suited to deal with color and scale changes, but it can also be successfully used in the presence of nonsegmented objects. Furthermore, H_{int} can be computed efficiently and adapted to search for partially occluded objects in images.

The effectiveness of histogram intersection as a similarity measure for color indexing raises the question of whether, or not, this measure can be adopted in kernel-based methods. We answer this question by finding explicitly the feature mapping after which histogram intersection is an inner product.

If we represent an histogram A with the P-dimensional binary vector $\mathbf{A}$ (with $P = N \times M$) defined as

$$\begin{aligned} \mathbf{A} = \Bigg(& \underbrace{\overbrace{1, \ldots, 1}^{A_1}, \overbrace{0, \ldots, 0}^{N-A_1}}_{\text{1st bin}}, \underbrace{\overbrace{1, \ldots, 1}^{A_2}, \overbrace{0, \ldots, 0}^{N-A_2}}_{\text{2nd bin}}, \ldots, \\ & \underbrace{\overbrace{1, \ldots, 1}^{A_M}, \overbrace{0, \ldots, 0}^{N-A_M}}_{M\text{th } bin} \Bigg) \end{aligned} \quad (14)$$

and, similarly, B with $\mathbf{B}$, it can be readily seen that $H_{\text{int}}(A, B)$ in (13) is equal to the standard inner product between the two

corresponding vectors $\mathbf{A}$ and $\mathbf{B}$, or in the notation of the previous section

$$H_{\text{int}}(A,B) = K_{\wedge}(A,B).$$

Thus, H_{int} is a Mercer's kernel and the binary vector in (14) describes explicitly the mapping $\mathbf{A} = \phi(A)$ between input and feature space.

A few comments are in order. First, we notice the redundant representation of the information content in (14). The P components of $\mathbf{A}$ are dependent *features* since the original histogram A is uniquely determined by the rightmost 1 in each of the MN-tuplets of components in (14).

Second, the assumption of dealing with images with the same number of pixels is unnecessary. In the case of images of different size, the same analysis can be obtained by normalizing the histogram areas and repeating the same construction of above on the resulting normalized histograms. Alternatively, it is sufficient to take P large enough (for example, equal to $\hat{N} \times M$ with $\hat{N}$ equal to the number of pixels of the largest image in the considered set).

Third, the generalization of this result to higher dimension like two-dimensional (2-D) images and three-dimensional (3-D) color space representations is straightforward.

Fourth, if we restrict our attention to the case in which the binary strings have the same number N of bits equal to 1, the close relation between histogram intersection and L^1 distance becomes apparent. Indeed, from the fact that for two strings of length P with N 1s each, $\hat{N}$ of which at the same location, we have $\mathbf{A} \cdot \mathbf{B} = \hat{N}$ and $\mathbf{A} \otimes \mathbf{B} = 2(N - \hat{N})$ it follows that

$$2\mathbf{A} \cdot \mathbf{B} + \mathbf{A} \otimes \mathbf{B} \equiv 2N. \tag{15}$$

Using (15) and the fact that the L^1 distance in input space is equal to the L^1 bit-wise distance in feature space, that is

$$\sum_{m=1}^{M} |A_m - B_m| = \sum_{p=1}^{P} |a_p - b_p|$$

we have that $H_{\text{int}}(A,B) = N - (1/2)||\mathbf{A} - \mathbf{B}||_{L^1}$. This leads to the conclusion that under the assumption of dealing with binary strings with the same number N of bits equal to 1, the L^1 distance is the distance *naturally* associated with histogram intersection [21], [22].

V. Hausdorff Kernel

In this section, we show that, by means of appropriate modifications, a similarity measure for gray-level images based on the notion of Hausdorff distance is a Mercer's kernel.

A. Matching Gray-Level Images With Hausdorff Distance

Let us start describing the essence of the similarity measure proposed in [20]. Without loss of generality and for ease of notation we consider the case of 1-D images.

Suppose we have two N pixel gray-level images, $\mathcal{A}$ and $\mathcal{B}$, of which we want to compute the degree of similarity. In order to compensate for small level changes or local transformations we define a neighborhood O of each pixel $i = 1, \ldots, N$ and evaluate the expression

$$h_{\mathcal{A}}(\mathcal{B}) = \sum_{i=1}^{N} U(\epsilon - |\mathcal{A}[i] - \mathcal{B}[s_i]|) \tag{16}$$

where U is the unit step function, and s_i denotes the pixel of $\mathcal{B}$ most similar to $\mathcal{A}[i]$ in the neighborhood $O(i)$ of i. For each pixel i, the corresponding term in the sum (16) equals 1 if $|\mathcal{A}[i] - \mathcal{B}[s_i]| \leq \epsilon$ (that is, if in the neighborhood $O(i)$ there exists a pixel s_i in which $\mathcal{B}$ differs by no more than ϵ from $\mathcal{A}[i]$) and 0 otherwise. The rationale behind (16) is to evaluate the degree of overlap of two images bypassing pixel-wise correspondences that do not take into account the effects of small variations and acquisition noise.

Unless the set O consists of the only element i, $h_{\mathcal{A}}(\mathcal{B}) \neq h_{\mathcal{B}}(\mathcal{A})$. Symmetry can be restored, for example, by taking the average

$$H(\mathcal{A},\mathcal{B}) = \frac{1}{2}(h_{\mathcal{A}}(\mathcal{B}) + h_{\mathcal{B}}(\mathcal{A})).$$

The quantity $H(\mathcal{A},\mathcal{B})$ can be computed in three steps.

1) *Expand* the two images $\mathcal{A}$ and $\mathcal{B}$ into 2-D binary matrices A and B, respectively, the second dimension being the gray value. For example, for A we write

$$A[i,j] = \begin{cases} 1, & \text{if } \mathcal{A}[i] = j \\ 0, & \text{otherwise.} \end{cases}$$

2) *Dilate* both matrices by growing their nonzero entries, i.e., by adding 1s in a neighborhood $\epsilon/2$ in the gray value dimension and half the linear size of O in the spatial dimension. Let A^D and B^D be the resulting 2-D dilated binary matrices.
3) *Compute* the size of the intersections between A and B^D, and B and A^D, and take the average of the two values. Thinking of the 2-D binary matrices as (binary) vectors in obvious notation we could also write

$$H(\mathcal{A},\mathcal{B}) = \frac{1}{2}(\mathbf{A} \cdot \mathbf{B}^D + \mathbf{B} \cdot \mathbf{A}^D). \tag{17}$$

Under appropriate conditions on the dilation sizes h is closely related to the partial directed Hausdorff distance between the binary matrices A and B thought of as 2-D point sets [23], [20]. As a similarity measure, the function H in (17) has been shown to have several interesting properties, like tolerance to small changes affecting images due to geometric transformations, viewing conditions, and noise, and robustness to occlusions [20].

B. Hausdorff-Based Mercer's Kernels

Here, again, we ask whether, or not, the similarity measure of above can be adopted as a kernel in statistical learning methods. For a fixed training set of images $\{\mathcal{A}_1, \ldots, \mathcal{A}_\ell\}$, an empirical answer to this question can be simply provided by checking the positive semidefiniteness of the $\ell \times \ell$ matrix

$$K_{ij} = H(\mathcal{A}_i, \mathcal{A}_j) = \frac{1}{2}\left(\mathbf{A}_i \cdot \mathbf{A}_j^D + \mathbf{A}_j \cdot \mathbf{A}_i^D\right).$$

$\mathbf{A}$ ↔	0	0	0	1	0	0	1	1	0	0	0
$\mathbf{A}^D$ ↔	0	1	1	1	1	1	1	1	1	1	0
$\mathbf{A}^S$ ↔	0	1/4	1/4	1	1/2	3/4	5/4	1	1/2	1/4	0

Fig. 1. Difference between $\mathbf{A}^D$ and $\mathbf{A}^S$. Given (top) the binary vector $\mathbf{A}$, (middle) the vector $\mathbf{A}^D$ is obtained by dilating each unit entry of $\mathbf{A}$ as specified in step 2) of Section V-A within a neighborhood of half-size 2. (Bottom) Vector $\mathbf{A}^S$ is obtained by convolving $\mathbf{A}$ with the stencil $S = (1/4, 1/4, 1, 1/4, 1/4)$. Notice that $\mathbf{A}^D$ is still a binary vector while $\mathbf{A}^S$ has rational entries that may also be greater than 1.

A simple counterexample shows that $H(\mathcal{A}, \mathcal{B})$ is not a Mercer's kernel. Consider the three one-pixel images $\mathcal{A}_1 = 1, \mathcal{A}_2 = 2$, $\mathcal{A}_3 = 3$, and $\epsilon = 1$. Explicit computation of the entries of the 3×3 matrix K gives

$$K = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

which, having negative determinant, is clearly not positive semidefinite.

In [24], H was used as a kernel for one class SV-based classification. The argument presented there to justify the convexity of the optimization problem was flawed: as we discussed in Section II, the fact that H is not a Mercer's kernel may lead to a nonconvex problem with multiple solutions. The very promising experimental results in [24] suggest that the local minimum obtained as the solution of the optimization problem was empirically close to the global minimum.

We now show that through an appropriate alteration of H, it is possible to obtain a Mercer's kernel without losing the attractive properties of Hausdorff-based similarity measures. The function H is not a Mercer's kernel since the dilation stage makes the off diagonal elements of the "kernel" matrix too large. Using the theorem of Section III-C, we see that if we redefine the dilation stage as a convolution with a 3-D stencil S of the type in expression (10) for all three dimensions, the newly defined quantity H_S

$$H_S(\mathbf{A}, \mathbf{B}) = \frac{1}{2}(\mathbf{A} \cdot \mathbf{B}^S + \mathbf{B} \cdot \mathbf{A}^S)$$

is an inner product [see (12)]. From the proof of the theorem, it follows that this result is made possible by 1) the *reduced amount* of dilation built in the convolution stage—as opposed to the original dilation stage—and 2) the linearity of the convolution operator. An example in the simpler 1-D case is shown in Fig. 1. In the 3-D case, the stencil is defined over a cuboid with the central entry equal to 1 and the off-center weights summing up to 1.

VI. Experimental Results

In this section, we describe the experiments we performed to corroborate the theoretical analysis presented in previous sections. We first consider two different image classification tasks addressed through histogram-based representations: indoor/outdoor classification and cityscape retrieval. Then we deal with the problem of 3-D object detection from single image with iconic representation.

Fig. 2. Samples of indoor images used in our experiments.

Fig. 3. Samples of outdoor images used in our experiments.

Fig. 4. Some examples in which the labeling is ambiguous (see text).

A. Histogram-Based Representation

Most of the images used for the problems of indoor/outdoor classification and cityscape retrieval were downloaded from the Web, by randomly crawling Web sites containing relevant images. A small percentage was composed by holiday photos of members of our group.

1) Indoor/Outdoor Image Classification: Deciding whether an image is a picture of an indoor or an outdoor scene is a typical example of understanding the *semantics* of the image content [25]. Its practical use ranges from scene understanding to

TABLE I
RRs FOR SVMs WITH DIFFERENT KERNELS ON THE INDOOR/OUTDOOR CLASSIFICATION PROBLEM IN THE HSV COLOR SPACE. THE RRs ARE OBTAINED BY AVERAGING OVER FOUR DIFFERENT SPLITS (THE INDICATED DEVIATIONS ARE STANDARD DEVIATIONS). IN THE CASE OF RBF KERNELS, ONLY THE PARAMETER VALUES LEADING TO THE BEST RRs ARE REPORTED

Kernel type	**Recognition rate** (%)		
	$8 \times 8 \times 8$	$15 \times 15 \times 15$	$20 \times 20 \times 20$
histogram intersection	69.6 ± 0.9	89.7 ± 0.3	72.0 ± 0.9
linear	70.6 ± 1.3	82.7 ± 1.2	68.2 ± 1.9
2-nd deg polynomial	69.9 ± 0.5	82.8 ± 1.1	68.7 ± 1.0
4-th deg polynomial	67.2 ± 0.7	81.9 ± 0.6	69.0 ± 0.5
Gaussian ($\sigma = 0.7$)	69.1 ± 0.6	83.5 ± 0.7	68.9 ± 1.1
Laplacian ($\sigma = 0.9$, $a = 1$)	71.2 ± 0.2	90.7 ± 1.0	73.2 ± 0.9
Laplacian ($\sigma = 10$, $a = 0.5$)	66.1 ± 0.8	90.8 ± 0.6	78.2 ± 0.9
Laplacian ($\sigma = 30$, $a = 0.25$)	66.5 ± 0.5	91.2 ± 0.7	79.1 ± 1.2
Sublinear ($\sigma = 50$, $a = 1$)	73.6 ± 0.4	89.7 ± 1.0	73.0 ± 1.6

TABLE II
RRs FOR SVMs WITH DIFFERENT KERNELS ON THE INDOOR/OUTDOOR CLASSIFICATION PROBLEM IN THE RGB COLOR SPACE. THE RRs ARE OBTAINED BY AVERAGING OVER FOUR DIFFERENT SPLITS (THE INDICATED DEVIATIONS ARE STANDARD DEVIATIONS). IN THE CASE OF RBF KERNELS, ONLY THE PARAMETER VALUES LEADING TO THE BEST RRs ARE REPORTED

Kernel type	**Recognition rate** (%)		
	$8 \times 8 \times 8$	$15 \times 15 \times 15$	$20 \times 20 \times 20$
histogram intersection	58.7 ± 2.2	88.4 ± 0.3	88.4 ± 1.1
linear	55.6 ± 1.0	82.6 ± 1.1	82.5 ± 1.7
2-nd deg polynomial	53.3 ± 1.7	83.6 ± 0.5	83.6 ± 1.8
4-th deg polynomial	50.4 ± 0.8	84.6 ± 0.8	84.5 ± 1.2
Gaussian ($\sigma = 0.5$)	50.8 ± 0.4	85.8 ± 1.2	84.3 ± 1.0
Laplacian ($\sigma = 1.0$, $a = 1$)	47.1 ± 0.7	90.4 ± 0.2	89.8 ± 1.2
Laplacian($\sigma = 10$, $a = 0.5$)	50.0 ± 0.3	90.2 ± 0.5	90.1 ± 0.9
Laplacian($\sigma = 30$, $a = 0.25$)	48.6 ± 0.1	89.8 ± 0.7	88.6 ± 0.4
Sublinear ($\sigma = 30$, $a = 1$)	48.6 ± 0.4	89.1 ± 0.3	89.2 ± 0.3

automatic image adjustment for high quality printout and intelligent film development.

For our experiments, we gathered a collection of 1392 outdoor and 681 indoor images. Figs. 2 and 3 show some examples of the used indoor and outdoor images, respectively. Fig. 4, instead, shows examples of ambiguous images, indoor images with natural outdoor light effects (often caused by the presence of windows), outdoor images taken at the sunset, or close-ups. The images of the dataset are not homogeneous in size, format, resolution, or acquisition device. Their typical size is in the range of 10^4–10^5 pixels. This problem can be naturally seen as a binary classification problem; we, therefore, use a support vector machine for binary classification. Since an important cue for this problem is color distribution [25], we decided to represent each image by means of a single, global color histogram and compare the performance of SVM for various kernels.

We built a training set of 400 indoor and 400 outdoor images, by randomly sampling the entire population. All remaining 1273 images (992 of which outdoor and 281 indoor) formed the test set.

The results we obtained on HSV and RGB color histograms comparing the histogram intersection kernel with common off-the-shelf kernels and the heavy-tail RBF kernels [10] defined as

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{1}{2\sigma^2}\sum_i |x_i^a - y_i^a|^b\right)$$

TABLE III
NUMBER OF SUPPORT VECTORS FROM DIFFERENT KERNELS, A TRAINING SET OF 800 ITEMS, $15 \times 15 \times 15$ BINS FOR BOTH HSV AND RGB

Kernel type	**RGB**	**HSV**
histogram int.	436	532
linear	372	419
polynomial (deg. 2)	355	395
polynomial (deg. 4)	342	370
Gaussian ($\sigma = 0.5$)	341	379
Gaussian ($\sigma = 0.7$)	364	396
Laplacian ($\sigma = 0.9$)	564	632
Laplacian ($\sigma = 1.0$)	544	615
Sublinear ($\sigma = 50$)	544	581

are shown in Tables I and II, respectively. As in [10], we call *Laplacian* kernels the heavy-tail RBF kernels with $b = 1$, and *Sublinear* kernels the ones with $b = 0.5$. For all RBF kernels, we trained the system varying the value of σ in a reasonably wide range. Here, we report the results obtained with the best performing σs for each type of kernel.

For both color spaces, in a first set of experiments, we built histograms with $15 \times 15 \times 15$ bins and then check the sensitivity to bin resolution considering coarser ($8 \times 8 \times 8$ bins) and finer color histograms ($20 \times 20 \times 20$ bins). For each kernel and each histogram resolution, the displayed recognition rates (RRs) were obtained by averaging the results over four different splits between training and test sets built according to the procedure of above and computing the associated standard

deviation. By inspecting Tables I and II, a number of conclusions can be drawn. First, histogram intersection and heavy-tail kernels behave better than off the shelf kernels at intermediate resolution ($15 \times 15 \times 15$ bins) for both color spaces. The results vary differently with respect to bin resolution for the two color spaces. In the RGB case, very good results are also obtained at the finer resolution (for a few kernels even better rates), while in the HSV case, there is a significant drop in performance for both coarser and finer resolution. At peak performances, however, the choice of the color space does not seem to be crucial, as reported in [10]. The reasonably small range of variation of the various standard deviations indicate that the obtained results are quite stable.

Second, heavy-tail kernels for a rather wide selection of parameters give slightly though consistently better results than histogram intersection (around ten out of 1000 images). This limited gain should be compared with the fact that histogram intersection has no parameter to tune and is much cheaper to compute. The issue of parameter estimation should not be neglected because the training sets were separated with no error by all kernels. This said, the heavy-tail kernels appear to be better suited to capture the sparsity of the color histograms.

Table III displays the support vector number for both HSV and RGB color spaces, respectively. From the entries of Table III, it is clear that the HSV representation leads to the selection of a higher number of support vectors. Furthermore, the number of support vectors for each color space is higher for the kernels performing best on the given test. The implications, if any, of these empirical findings need to be investigated. In the case of HSV space, we also performed experiments with an unbalanced sampling (different number of bins per channel). In the case of indoor/outdoor classification, the quantity of light and saturation seemed to be more important than the shade of colors; for this reason, we built histograms with only four bins for the hue (instead of 15). The RRs drop about 1% for all kernels, with respect to the $15 \times 15 \times 15$ case, but this choice is an interesting compromise allowing us to still achieve very good results with a significant reduction in the dimensionality of the input space.

We conclude by observing that the results obtained using appropriate kernels (both histogram intersection and heavy-tail kernels) are very close to those reported in [25], in which a fairly complex classification procedure was proposed and tested on a different database reaching an RR slightly better than 90%. Overall, it appears that the learning from examples approach is able to exploit the fact that color distribution captures an essential aspect of the indoor/outdoor classification problem.

2) Cityscape Retrieval: The problem of finding cityscapes in an image dataset[3] amounts to finding a method for understanding whether an image is a picture taken from a distance sufficient to obtain a panoramic view of a city: Skylines are well-known examples of cityscapes.

For this set of experiments, we collected a dataset of 803 positive examples consisting of open city views and skylines (some examples of which are shown in Fig. 5). Somewhat arbitrarily,

[3]The relevance of this problem to news agencies has been brought to our attention by Reuters.

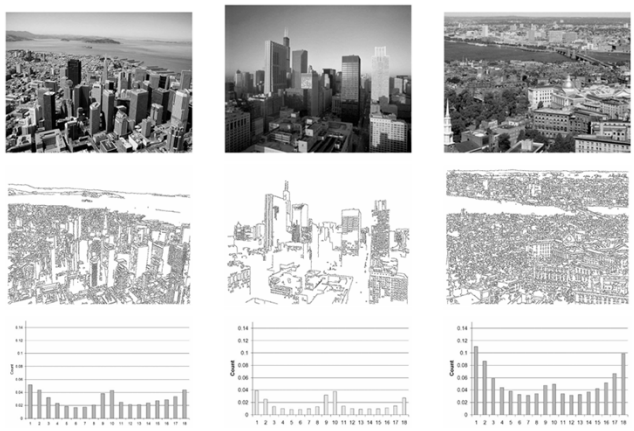

Fig. 5. (Top) Samples of cityscapes, (middle) the detected edge points, (bottom) and the histograms of edge directions.

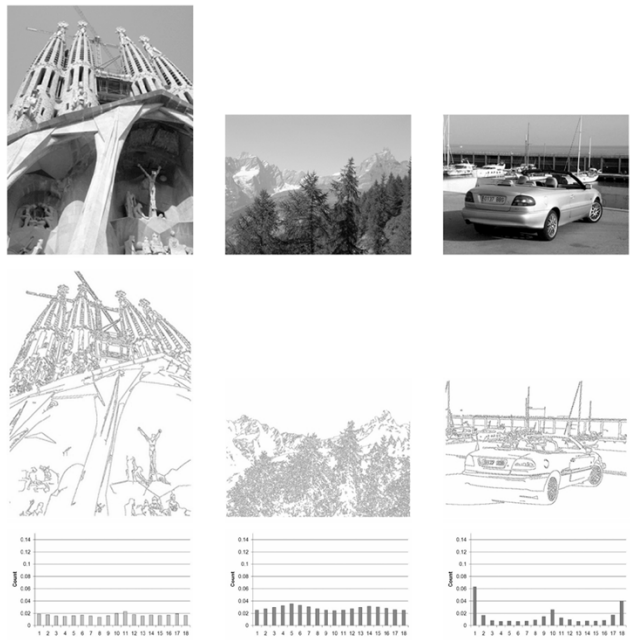

Fig. 6. (Top) Samples of noncityscapes, (middle) the detected edge points, (bottom) and the histograms of edge directions.

we decided to sample the space of *noncityscapes* gathering 849 examples of city details (like street corners, buildings, etc.), different sorts of man-made objects, people, and landscapes (see Fig. 6 for some examples). An important cue is the geometry of the scene: a possible approach is to look for specific geometric features, like straight lines or polygons. In the attempt to keep the preprocessing stage to a minimum and investigate the relevance of using appropriate kernels in the statistical learning framework, we resort to using a binary SVM classifier and representing image information with edge direction histograms. We use the Canny edge detector to extract edges and estimate their orientation. Histograms were obtained dividing the range between 0° and 180° in ten equally spaced intervals and adding a further bin for nonedge pixels.

As in the previous set of experiments, we formed a training set of 800 images (400 cityscapes and 400 noncityscapes) randomly sampling the image data set and using the remaining 852

TABLE IV
ACCURACY OF THE CITYSCAPE RETRIEVAL SYSTEM, FOR DIFFERENT KERNELS AND TWO DIFFERENT IMAGE FILTERINGS. IN THE CASE OF RBF KERNELS, ONLY THE PARAMETER VALUES LEADING TO THE BEST RECOGNITION RATES ARE REPORTED

	$\sigma_e = 1.0$		$\sigma_e = 2.0$	
Kernel type	**Precision** (%)	**Recall** (%)	**Precision** (%)	**Recall** (%)
histogram intersection	85.1 ± 1.7	90.1 ± 1.7	82.2 ± 1.4	87.1 ± 3.2
linear	86.0 ± 0.0	82.8 ± 1.9	83.6 ± 0.3	80.2 ± 2.6
2-nd deg polynomial	84.2 ± 0.7	85.9 ± 4.6	83.4 ± 1.0	83.8 ± 2.7
4-rd deg polynomial	82.3 ± 0.3	90.0 ± 2.7	82.6 ± 1.9	84.2 ± 2.1
Gaussian RBF ($\sigma = 1.0$)	83.5 ± 2.8	79.7 ± 6.7	84.0 ± 0.6	79.5 ± 2.6
Laplacian RBF ($\sigma = 3.0$)	84.4 ± 1.4	83.0 ± 1.6	83.1 ± 1.0	82.3 ± 1.2

images of the entire dataset as the test set (403 cityscapes and 449 noncityscapes).

Table IV shows the RRs obtained using different kernels and two different widths of the low-pass filter used for edge detection. In the case of RBF kernels, only the values leading to the best RRs are included.

The results are displayed in terms of *precision*, the proportion of cityscapes within all images retrieved by the system, and *recall*, the proportion of cityscapes retrieved by the system within all cityscapes in the test set. The precision-recall evaluations is popular in the world of retrieval: The measure of precision is imporant for retrieval purposes as it describes the proportion of retrieved items that are correct for the query; the measure of recall is important for classification purposes, since it gives the percentage of positive items detected in a validation data set. In a good classifier, the two quantities should be high and possibly well balanced (otherwise further tuning is required).

As it can be seen by inspecting Table IV, the histogram intersection kernel in this case appears to perform best. The effectiveness of heavy-tail kernels seem rather limited, if at all. The combination of parameters leading to the best results are displayed in the table.

This quite different behavior is likely to be due to the very different nature of the inputs between this and the previous set of experiments. Here, histograms live in a space of 11 dimensions and, since the typical image size is 10^3–10^4 pixels, are dense. As before, the model selection is not straightforward, given the fact that for all kernels we obtained perfect separation of the training set for $C = 100$. Once more, the advantage of using a kernel which does not depend on free parameter is clear. Concluding, we observe that these results, which represent a substantial improvement with respect to the same approach with co-occurrence matrices instead of histograms of edge direction [26], are very similar to those reported in [27], possibly the state-of-the-art on the subject. There is little doubt that better results could be obtained by employing more sophisticated representations.

B. Iconic Representation

To evaluate the effectiveness of the Hausdorff-based kernel described in Section V, we performed some experiments with a view-based identification system. The idea is to train a system from positive examples only to identify a 3-D object from a single image. Following an appearance-based approach, the training set consists of a collection of views of a certain object taken by a camera moving around it. At run time the system must answer the question of whether the current image, or a part of it, is a view of the same object. The object is a toy duck to be detected in a cluttered environment. An important component of this system is the technique used for image matching. Fig. 7 shows example frames belonging to the training sequence. The collection of views is built using a semiautomatic tracking stage, the purpose of which is to obtain a crude form of foreground/background segmentation. A rectangular region containing the object of interest is identified in the first frame and used as a template. The region is then tracked through the sequence maximizing in each frame the matching score of (17)—with tolerance on the gray levels $\epsilon = 3$ and neighborhood $N_p = 3 \times 3$. If the maximized matching score falls below a given threshold, a new rectangular region, or template, is drawn around the object of interest in the current frame and the procedure is restarted. Fig. 8 shows the results of the tracking after initialization with the template shown in the first frame of each figure, respectively. It can be easily checked that the object is not always centered since the threshold determining the need for a new initialization of the tracking procedure has been set to a small number to minimize manual intervention. Each image in the segmented sequence is resized to 69×47 (therefore, the data live in a space of 3243 dimensions).

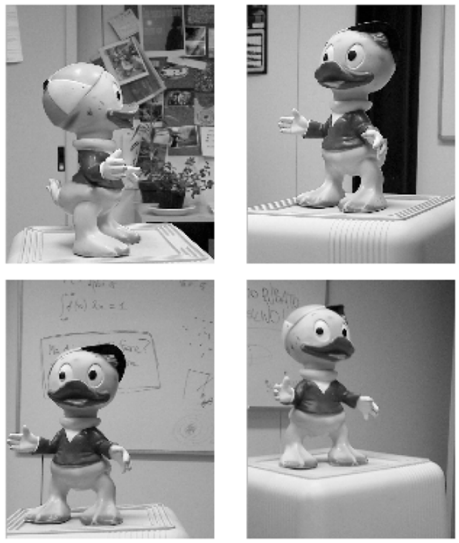

Fig. 7. Example frames from the original toy duck sequence.

In our experiments, we train a one-class SVM [16] on 460 images taken from a single image sequence. We test the system on two different image sequences of the same object (a total of 660

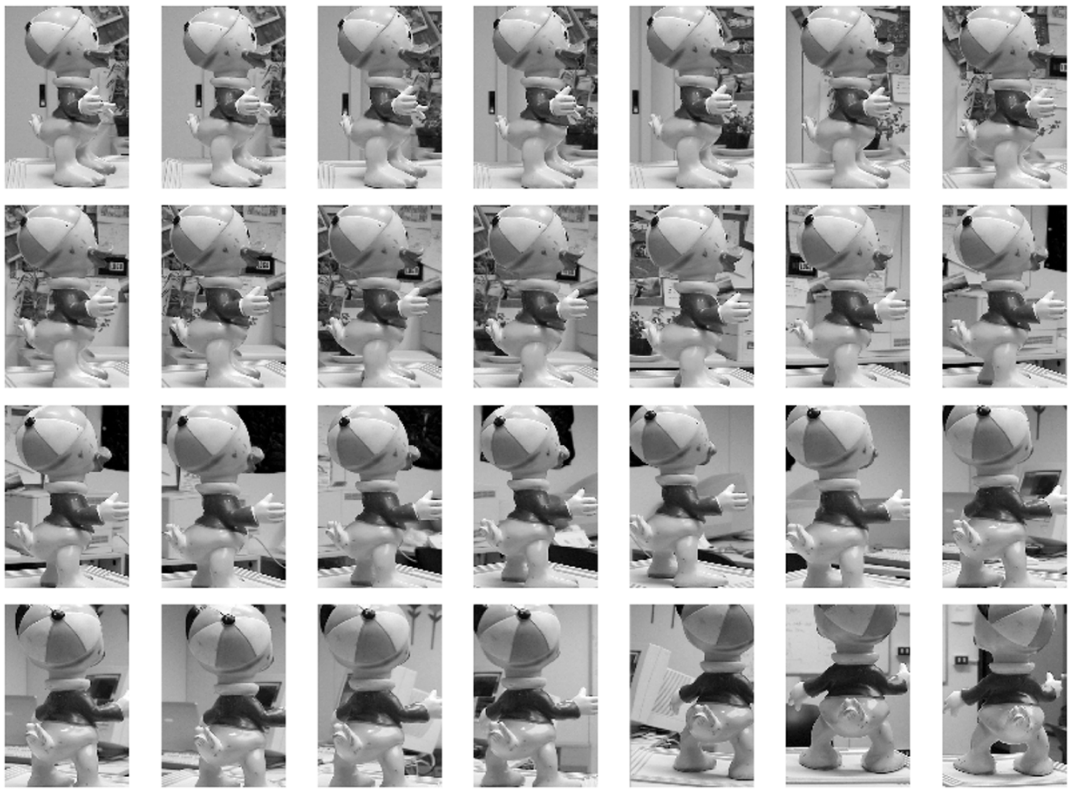

Fig. 8. Tracking results after initialization with the template shown in the first frame. See text for a description of the tracking procedure.

images) and 1360 images of negative examples. The positive test data have been acquired in different days by different people. To avoid searching across scale, in the acquisition stage, the apparent size of the object of interest throughout the sequences was kept approximately constant. The negative test data contain other views of the same environment, including similar objects (e.g., other toys) at roughly the same scale. Clearly, searching across scale can not be avoided when developing a practical detection system.

The results of the experiments performed are shown by means of receiver operating characteristic (ROC) curves. Each point of a ROC curve represents a pair consisting of the false-alarm rate and the hit rate of the system obtained by varying the radius of the sphere in feature space. The system efficiency can be evaluated by the growth rate of its ROC curve, and for a given false-alarm rate, the better system will be the one with a higher hit probability. The overall performance of a system can be measured by the equal error rate (e.e.r.) representing the best compromise between false alarms and hit rate. Since the results obtained seem hardly affected by changing the regularization parameter C of SVMs in the range 0.1–0.9, we fix $C = 0.2$.

The ROC curves in Figs. 9–11 show the performances obtained for various kernels on the identification problem described above. Fig. 9 compares the results of polynomial kernels of various degrees. It shows immediately that more complex polynomials do not raise recognition accuracy. Fig. 10 shows the results obtained with Gaussian RBF kernels of different σs. As illustrated by the figure, it is not easy to find a range of σ leading to acceptable results. Fig. 11 compares the performances of the modified Hausdorff kernel

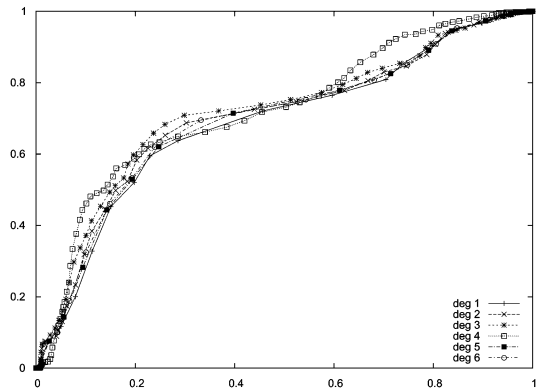

Fig. 9. ROC curves comparing various polynomial kernels for the toy duck experiment (see text).

for different 3-D stencils.The curve identified by the triple $x\ y\ z$ corresponds to a stencil defined over a cuboid of side $(2x+1) \times (2y+1) \times (2z+1)$.

Concluding, we also checked the performance of nearest-neighbors techniques, using the distance measures naturally defined by the various kernels. The obtained results are invariably poor with an e.e.r. in the range of 40%.

VII. Conclusion

In this paper, we dealt with the problem of engineering kernels for images. We represented images with binary strings and discussed bitwise manipulations obtained using logical operators and convolution with nonbinary stencils. From the theoretical viewpoint, we used our analysis to show that histogram in-

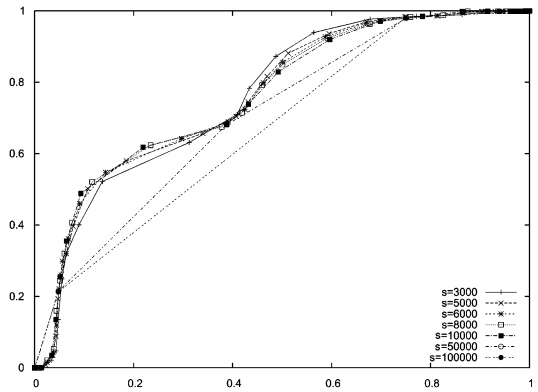

Fig. 10. ROC curves comparing various Gaussian RBF kernels for the toy duck experiment (see text).

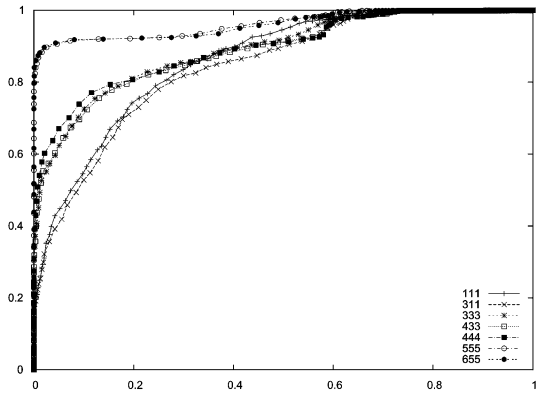

Fig. 11. ROC curves comparing Hausdorff-based Mercer's kernel for various stencils for the toy duck experiment (see text).

tersection is a Mercer's kernel and determine the modifications under which a similarity measure based on the notion of Hausdorff distance is a Mercer's kernel. Extensive experimental results support the tenet that choosing an appropriate kernel leads to the construction of more effective trainable systems. Current work on the topic of kernel engineering focuses on extending the presented analysis to binary strings of different length and studying the relation with string kernels.

ACKNOWLEDGMENT

The authors would like to thank J. Shawe-Taylor, Y. Singer, and E. Franceschi for useful discussions.

REFERENCES

[1] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[2] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[3] M. Pontil and A. Verri, "Support vector machines for 3-D object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 6, pp. 637–646, Jun. 1998.
[4] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *Int. J. Comput. Vis.*, vol. 38, no. 1, pp. 15–33, 2000.
[5] G. Guodong, S. Li, and C. Kapluk, "Face recognition by support vector machines," in *Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition*, Mar. 2000, pp. 196–201.
[6] K. Jonsson, J. Matas, J. Kittler, and Y. Li, "Learning support vectors for face verification and recognition," in *Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition*, Mar. 2000, pp. 28–30.
[7] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: Global versus component-based approach," in *Proc. ICCV*, 2001, pp. 688–694.
[8] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 23, no. 4, pp. 349–361, Apr. 2001.
[9] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik, "Prior knowledge in support vector kernels," in *Proc. Advances in Neural Information Processing Systems*, vol. 10, 1998, pp. 640–646.
[10] O. Chapelle, P. Haffner, and V. Vapnik, SVMs for histogram-based image classification, in *IEEE Trans. Neural Netw.*, to be published.
[11] F. Jing, M. Li, H. Zhang, and B. Zhang, "Support vector machines for region-based image retrieval," in *Proc. IEEE Int. Conf. Multimedia and Expo.*, vol. 2, Jul. 2003, pp. II-21–II-4.
[12] S. Belongie, C. Fowlkes, F. Chung, and J. Malik, "Spectral partitioning with indefinite kernels using the nyström extention," in *Proc. ECCV*, Copenhagen, Denmark, May 2002.
[13] C. Wallraven, B. Caputo, and A. Graf, "Recognition with local features: The kernel recipe," in *Proc. Int. Conf. Computer Vision*, 2003, pp. 237–264.
[14] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Adv. Comput. Math.*, vol. 13, pp. 1–50, 2000.
[15] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Wiley, 1995.
[16] D. M. J. Tax and R. Duin, "Uniform object generation for optimizing one-class classifiers," *J. Mach. Learning Res.*, vol. 2, pp. 155–173, 2002.
[17] R. Courant and D. Hilbert, *Methods of Mathematical Physics*. London, U.K.: Interscience, 1962, vol. 2.
[18] T. Poggio, S. Mukherjee, R. Rifkin, A. Rakhlin, and A. Verri, "b," in *Proc. Conf. Uncertainty in Geometric Computations*, 2001, pp. 22–28.
[19] K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.
[20] F. Odone, E. Trucco, and A. Verri, "General purpose matching of gray level arbitrary images," in *Proc. 4th Int. Workshop on Visual Forms*, 2001, pp. 573–582.
[21] M. J. Swain and D. H. Ballard, "Color indexing," in *Proc. IJCV*, vol. 7, 1991, pp. 11–32.
[22] M. J. Swain, "Interactive indexing into image databases," in *Proc. Storage and Retrieval for Image and Video Databases*, 1993, pp. 95–103.
[23] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.
[24] A. Barla, F. Odone, and A. Verri, "Hausdorff kernel for 3-D object acquisition and detection," in *Proc. ECCV*, 2002, pp. 20–33.
[25] M. Szummer and R. W. Picard, "Indoor-outdoor image classification," in *Proc. IEEE Int. Workshop on Content-Based Access of Image and Video Databases*, Jan. 1998, pp. 42–51.
[26] A. Barla, F. Odone, and A. Verri, "Old fashioned state-of-the-art image classification," in *Proc. Int. Conf. Image Analysis and Processing*, Mantova, Italy, 2003, pp. 566–571.
[27] A. K. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern Recognit.*, vol. 29, pp. 1233–1244, Aug. 1996.

Francesca Odone received the Laurea degree in information science and the Ph.D. in computer science from the University of Genova, Genova, Italy, in 1997 and 2002, respectively.

She was with Heriot-Watt University, Edinburgh, U.K., as a Research Associate in 1997 and as a visiting Ph.D. student in 1999. Since 2002, she has been a Researcher for the Istituto Nazionale di Fisica della Materia, working at the Department of Computer and Information Science, University of Genova. She has published papers on computer vision applied to automation, motion analysis, image matching, image classification, and view-based object recognition. Her present research focuses on statistical learning and its application to computer vision and image understanging problems.

Annalisa Barla received the Laurea degree in physics from the University of Genova, Genova, Italy, in 2001. She is currently pursuing the Ph.D. degree in computer science at the Department of Computer and Information Science, University of Genova.

Her main interests in research are connected with statistical learning applied to image understanding problems, such as content-based image retrieval and classification. She has published papers on image representation and kernel engineering for image classification.

Alessandro Verri received the Laurea and Ph.D. degrees in physics from the University of Genova, Genova, Italy, in 1984 and 1989, respectively.

Since 1989, he has been with the University of Genova, where he is a Professor with the Department of Computer and Information Science. He has been a Visiting Scientist and Professor at the Massachusetts Insitute of Technology, Cambridge; INRIA, Rennes, France; ICSI, Berkeley, CA; and Heriot-Watt University, Edinburgh, U.K. He has published nearly 60 papers on stereopsis, motion analysis in natural- and machine-vision systems, shape representation and recognition, pattern recognition, and 3-D object recognition. He is also the coauthor of a textbook on computer vision with Dr. E. Trucco. Currently, he is interested in the mathematical and computational aspects of computer vision and the statistical learning theory and he is leading a number of applied projects in the development of computer vision-based solutions to industrial problems. He has been on the program committees of major international conferences in the areas of image processing and computer vision and serves as a referee of several leading journals in the field.