Electronic Letters on Computer Vision and Image Analysis 7(2):96-109, 2008

Class Specific Object Recognition using Kernel Gibbs Distributions

Barbara Caputo

IDIAP Research Institute, Centre Du Parc, rue Marconi 19, 1920 Martigny Switzerland
Received 9th October 2007, Revised 26th May 2008; accepted 10th July 2008

Abstract

Feature selection is crucial for effective object recognition. The subject has been vastly investigated in the literature, with approaches spanning from heuristic choices to statistical methods, to integration of multiple cues. For all these techniques the final result is a common feature representation for all the considered object classes. In this paper we take a completely different approach, using class specific features. Our method consists of a probabilistic classifier that allows us to use separate feature vectors, selected specifically for each class. We obtain this result by extending previous work on Class Specific Classifiers and Kernel Gibbs distributions. The resulting method, that we call Kernel-Class Specific Classifier, allows us to use a different kernel for each object class by learning it. We present experiments of increasing level of difficulty, showing the power of our approach.

Key Words: object recognition, machine vision, statistical pattern analysis

1 Introduction

Object recognition is a key topic in computer vision, where a consistent amount of research has been devoted to developing effective features (see for instance [16, 20, 18, 4, 14] and many others). Thanks to these efforts, today it is possible to represent objects effectively by using color information [20, 14], global or local textural information [16, 18, 19], shape contour [4, 15], and so on. Still, it is not clear which features should be used for a given task. Consider for example the problem of classifying a vegetable as a member of 5 possible classes: tomatoes, carrots, zucchini, onions and pumpkins. If we represent these classes using the color representation, tomatoes will be identified with no ambiguities, but carrots and pumpkins will be mixed. If we choose instead a shape representation for all classes, tomatoes and onions could be confused, while pumpkins would be unambiguously identified; and so on. The dilemma of how to choose optimal features has been tackled mainly in two ways in the literature:

1. a very popular solution is to choose a single feature representation. This choice can be done with the help of some feature selection technique [22, 11], or (more often) on the basis of some prior knowledge or assumption on the nature of the task [20, 16, 18];

Correspondence to: <bcaputo@idiap.ch>

Recommended for acceptance by João Manuel R. S. Tavares and Renato Natal Jorge
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

2. another possibility is using multiple features, whether combined via a voting scheme [12], whether via a probabilistic model [7], whether defining a new feature representation that integrates visual information traditionally represented by different set of features [14].

Note that, in both cases, the approach will use the same features for each object. Still, it is fair to say that, for each class, there is at least one representation that captures at best its "essence", and thus makes it easily distinguishable. This representation can be different for different classes. Assuming we can determine it, the ideal solution would be to use class-specific features.

Baggenstoss et al. [2, 3] proposed a *feature-based Class Specific Classifier* (CSC). It allows the use of different features for different classes in a Bayesian framework. This result is obtained introducing a common reference hypothesis class and using results of statistical theory [2, 3]. An open point for CSCs is how to choose features for each class. The strategy to choose class specific features will be winning as long as the chosen features are the right ones, according to the need of that class. For most applications, the kind of features that can be used is, a priori, huge. Choices are usually made heuristically, and the truth is that, even when the performance of the final classifier is good, we cannot be sure that it wouldn't improve with another set of features. How to choose features for a given set of classes is an open problem for visual recognition [5, 9]. For CSCs, which base their power on the possibility to choose several sets of features for different classes, the problem is more relevant.

In the last decades, *kernel methods* have been proposed as an alternative solution to the feature selection problem. [17]. Kernel methods apply to every algorithm that depends on the scalar product between data. They replace the scalar product with a *kernel function*, which can be interpreted as the scalar product between the original data in a higher dimensional space. This space is reached via a non linear mapping, that replaces the feature extraction step. We stress that here the kernel function -and not the mapping -is explicitly known. Due to theoretical constraints, the functional form of kernels is known and limited [17]. Thus, the number of choices is small compared to the choice of a set of features, although the criteria is still mostly heuristic.

In this paper we propose combining CSC with kernel methods via SG-MRFs [6, 7], a class of Gibbs distributions that use energy functions inspired by results of SG theory [1] via kernel functions (mainly Gaussian kernels, [17]) . The use of SG-MRF in a CSC carries many advantages. First, it does allow to use the power of kernel functions for classification purposes, still leaving open the possibility to use different sets of features. Second, the class of possible kernel functions is determined a priori by theoretical constraints. As the choice of kernels is limited, it does not become ungovernable. At the same time is wide enough to reasonably guarantee the possibility to tailor each kernel according to each class needs. For each class, the kernel is *learned* by the training data with a leave-one-out strategy. We call this new method Kernel-Class Specific Classifier (K-CSC).

We tested K-CSC in the domain of object recognition, performing experiments of increasing difficulty. We benchmarked our method against SG-MRFs and Support Vector Machines, respectively a probabilistic and discriminative kernel method. Results clearly show the value of our approach. The rest of the paper is organized as follows: after a brief review of the probabilistic approach to appearance based object recognition (section 2), we describe the theory behind the Class Specific Classifier (section 3) and Spin Glass-Markov Random Fields (section 4). In section 5 we derive our new algorithm, and section 6 reports our experimental evaluation. The paper concludes with a summary discussion and possible avenues for future research.

2 Probabilistic Appearance Based Object Recognition

We are here concerned with any statistical method where it is needed to estimate the probability density function of the input data. We will focus on an important example of such methods: the so-called $\mathcal{K}$-ary classifier. Let $H_k, k = 1, \ldots \mathcal{K}$ be $\mathcal{K}$ different classes or statistical hypotheses: given a data sample $\boldsymbol{x}$, produced by one of $\mathcal{K}$ possible classes, our goal is to classify $\boldsymbol{x}$ as a sample from H_{k^*}, one of the H_k classes. It is well known that the optimal classifier, which results in the lowest probability of error is the Maximum A Posteriori (MAP)

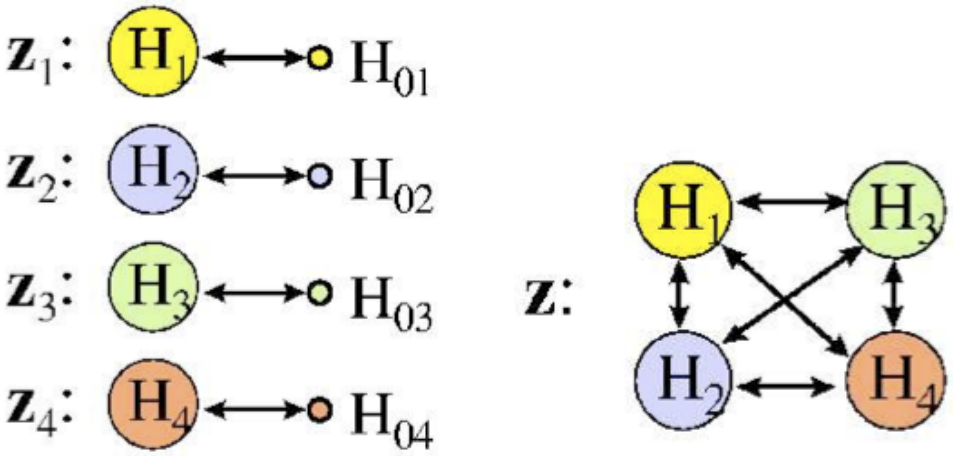

Figure 1: Feature extraction for different classes: on left, for each feature is extracted a specific feature; on right, a common feature is extracted for all the considered classes.

classifier [9]:

$$k^* = \underset{k=1,\ldots,\mathcal{K}}{\operatorname{argmax}} P(H_k|\boldsymbol{x}) = \underset{k=1,\ldots,\mathcal{K}}{\operatorname{argmax}} \{P(\boldsymbol{x}|H_k)P(H_k)\} \tag{1}$$

using Bayes rule, where $P(\boldsymbol{x}|H_k)$ are the Likelihood Functions (LFs) and $P(H_k)$ are the prior probabilities of the classes. Assuming that $P(H_k)$ are constant and equal, the Bayes classifier simplifies to

$$k^* = \underset{k=1,\ldots,\mathcal{K}}{\operatorname{argmax}} P(\boldsymbol{x}|H_k).$$

In well defined applications the LFs can be written down [9]. Still, most problems of practical interest involve the classification of patterns which either are unknown or do not obey statistical models that can be easily described. This is the case for visual object recognition in unconstrained settings. As a result, the LFs have traditionally had to be learned from data samples.

3 Class Specific Classifier

The high dimension of the raw data usually precludes doing any kind of non-parametric PDF estimation. Therefore, the usual approach extracts a small number of information-bearing statistic, called *features* (Figure 1). Let $\boldsymbol{z} = T(\boldsymbol{x})$ be such a set of features. The Bayesian classifier based on $\boldsymbol{z}$ is

$$k^* = \underset{k=1,\ldots,\mathcal{K}}{\operatorname{argmax}} P(\boldsymbol{z}|H_k). \tag{2}$$

Thus, the features replace the raw data. The hidden implication here is that $\boldsymbol{z}$ is a sufficient statistic for the classification problem:

$$P(\boldsymbol{x}|H_k) = g(T(\boldsymbol{x})|H_k)h(\boldsymbol{x}), k = 1,\ldots,\mathcal{K}. \tag{3}$$

This is the Neyman-Fisher factorization theorem [9]. Conceptually, learning of LFs is simple. The difficulty in PDF estimation is that as the dimension of $\boldsymbol{z}$ increases, the complexity of the problem increases exponentially. Indeed, it has been shown that given that the PDF meets certain smoothness assumptions, the amount of training

data required for nonparametric estimators* rises exponentially with the dimension [5]; the rapid increase in complexity of systems has been termed the *curse of dimensionality*.

Dimensionality reduction is often a necessary first step to PDF estimation and is the subject of much research currently and over the past decades [17, 5, 9]. The trouble is that projecting the data to a low-dimensional feature space requires knowing the nature or the structure of the mechanism; but knowing the structure requires knowing something about the PDF; and yet learning the PDF requires first knowing the structure. This frustrating dilemma has given rise to a seemingly infinite number of ingenious methods, each based on its own implicit assumptions. Various approaches include feature selection [9], projection pursuit [5] and independence grouping [9, 5]. Several other methods are based on projection of the feature vectors onto lower dimensional subspaces. A very large percentage of these methods can be characterized as a three-step process: (1) collect a number of information-bearing features, (2) reduce the dimension of this feature set, (3) estimate the PDF of the reduced-dimension feature-set conditioned on each data class.

In all of these methods, there is an implicit approximation which limits the theoretical performance. For example, subspace methods project the data to low-dimensional subspaces, which may be inappropriate in some problems. A visual example is a 2-dimensional plane in which one data class is distributed in a ring enclosing the samples of a second data class. Any attempt to project the samples to a lower-dimensional space (a line) will fail to preserve the class separation. Only a non-linear transformation (i.e. to polar coordinates) resolves the classes. Unfortunately, the correct non-linear transformation cannot be easily discovered at high dimension, such as in the simple example just described. Another common characteristic of these methods is that the relationship between the features and the raw data is forgotten. The class-specific method differs in both respect. Dimension reduction is accomplished at the theoretical level, before any practical approximations are made, and the relationship to the input data is preserved.

It was recognized for some time [9] that the $\mathcal{K}$-ary classifier (1) can be constructed by knowing only the likelihood ratios

$$\frac{P(\boldsymbol{x}|H_2)}{P(\boldsymbol{x}|H_1)}, \frac{P(\boldsymbol{x}|H_3)}{P(\boldsymbol{x}|H_1)}, \ldots, \frac{P(\boldsymbol{x}|H_M)}{P(\boldsymbol{x}|H_1)}.$$

Thus, these likelihood ratios are *sufficient*. Van Trees recognized [9, 5] that an additional class, H_0, the "dummy" class, can be used in the denominator:

$$\underset{k=1,\ldots,\mathcal{K}}{\operatorname{argmax}} \frac{P(\boldsymbol{x}|H_k)}{P(\boldsymbol{x}|H_0)}. \tag{4}$$

The well known corollary of the Neyman-Fisher factorization theorem (3) is that any likelihood ratio is invariant when written in terms of a sufficient statistic. Thus, if $\boldsymbol{z}$ is a sufficient statistic,

$$\frac{P(\boldsymbol{x}|H_j)}{P(\boldsymbol{x}|H_k)} = \frac{P(\boldsymbol{z}|H_j)}{P(\boldsymbol{z}|H_k)}. \tag{5}$$

The $\mathcal{K}$-ary classifier (1) can be constructed by knowing only the likelihood ratios; moreover, it is possible to use in the denominator an additional class, H_0:

$$k^* = \underset{k=1,\ldots,\mathcal{K}}{\operatorname{argmax}} \frac{P(\boldsymbol{x}|H_k)}{P(\boldsymbol{x}|H_0)}. \tag{6}$$

Now, note that the likelihood ratio doesn't change when we take two different mapping $z_1 = T_1(\boldsymbol{x}), z_2 = T_2(\boldsymbol{x})$:

$$\frac{P(\boldsymbol{x}|H_k)}{P(\boldsymbol{x}|H_0)} = \frac{P(\boldsymbol{z}_1|H_k)}{P(\boldsymbol{z}_1|H_0)} = \frac{P(\boldsymbol{z}_2|H_k)}{P(\boldsymbol{z}_2|H_0)}.$$

*By non-parametric, here we mean that we do not know the parametric form of the PDF, so we have to assume a standard form such as Gaussian mixture, then estimate the parameters of this assumed model with the understanding that the model we assume is just an approximation.

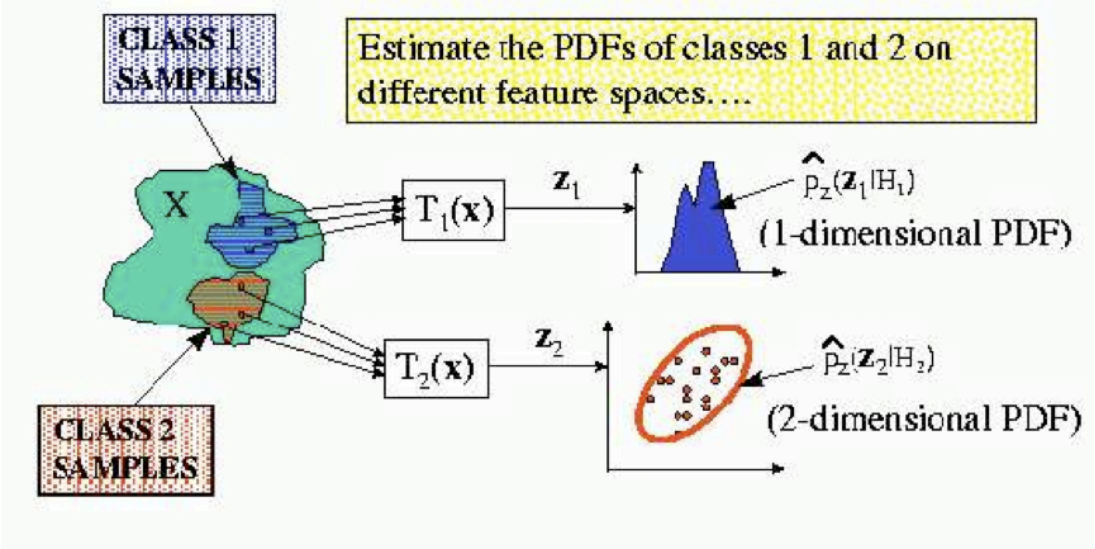

Figure 2: Class Specific Classifier: the PDF is estimated on a different feature space, for each class.

And thus we may write the Bayes classifier as [2, 3]

$$k^* = \underset{k=1,\ldots,\mathcal{K}}{\operatorname{argmax}} \frac{P(\boldsymbol{z}_k|H_k)}{P(\boldsymbol{z}_k|H_0)}, \tag{7}$$

where $\boldsymbol{z}_k = T_k(\boldsymbol{x}), 1 \leq k \leq \mathcal{K}$ are feature transformations that depend on the class being tested, thus they are *class specific* features. This is the *feature-based Class Specific Classifier* (CSC [2], Figure 2). CSCs major advantage is that they allow us to use different features for each class. A serious problem is that they do not provide any criteria on how to choose the optimal set of features for each class. Note that eq (7) holds only when $\boldsymbol{z}$ is a sufficient statistic. When this is not the case, eq (5) becomes:

$$\frac{P(\boldsymbol{x}|H_j)}{P(\boldsymbol{x}|H_k)} \approx \frac{P(\boldsymbol{z}|H_j)}{P(\boldsymbol{z}|H_k)},$$

and therefore eq (7) becomes

$$k^* \approx \underset{k=1,\ldots,\mathcal{K}}{\operatorname{argmax}} \frac{P(\boldsymbol{z}_k|H_k)}{P(\boldsymbol{z}_k|H_0)},$$

i.e. the class label computed by the CSC is an estimate of the true class label. Note that this is true not only for eq (7) but also for eq (6), the classic $\mathcal{K}$-ary classifier. This is due to the intrinsic and unavoidable uncertainty in choosing the features $\boldsymbol{z}$. In other words, whenever features are used, it is never known if they are a sufficient statistic or not. This leads to an estimate of the true class label, that does not depend on the CSC, but that holds for every probabilistic classifier.

4 Spin Glass-Markov Random Fields

This section introduces Spin Glass-Markov Random Fields (SG-MRFs), a class of kernel Gibbs distributions that connects SG-like energy functions (mainly the Hopfield one [1]) with Gibbs distributions via a nonlinear kernel mapping. An important characteristic of SG-MRF is that the functional form of the Gibbs distribution is written down in a parametric form as a function of Mercer kernels. As we will show in section 5, this property will lead us to the derivation of K-CSC. In the following we just briefly introduce SG-MRF. The interested reader will find the full derivation and a comprehensive discussion in [7]. Consider $\mathcal{K}$ different object classes

$\Omega_1, \Omega_2, \ldots, \Omega_{\mathcal{K}}$, and for each class a set of data samples $\omega_k = \{\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \ldots, \boldsymbol{x}_{n_k}^k\}, k = 1, \ldots \mathcal{K}$. The SG-MRF probability distribution is given by

$$P(\boldsymbol{x}|\Omega_k) = \frac{1}{Z} \exp\left(\frac{1}{N} \sum_{\mu=1}^{p_k} \left[K_{d-G}(\boldsymbol{x}, \tilde{\boldsymbol{x}}^{(\mu)})\right]^2\right) \tag{8}$$

with K_{d-G} generalized Gaussian kernel

$$K_{d-G} = \exp\{-\rho d_{a,b}(\boldsymbol{x}, \boldsymbol{y})\}, \qquad d_{a,b} = \sum_{i=1}^{m} |x_i^a - y_i^a|^b. \tag{9}$$

$\{\tilde{\boldsymbol{x}}^\mu\}_{\mu=1}^{p_k \equiv n_k} = \{\boldsymbol{x}_1^k, \ldots, \boldsymbol{x}_{n_k}^k\}$ are a set of vectors selected (according to a chosen ansatz, [7]) from the training data that we call prototypes. The number of prototypes per class must be finite, and they must satisfy the condition $K(\tilde{\boldsymbol{x}}^i, \tilde{\boldsymbol{x}}^j) = 0$, for all $(i, j) = 1, ..n_k$ and $i \neq j$. Note that SG-MRFs can be defined on features and on raw pixel data. The sites are fully connected, which ends in learning the neighborhood system from the training data instead of choosing it heuristically. Another key characteristic of the model is that in SG-MRF the functional form of the energy is given by construction.

The next section will sketches the theoretical derivation of the model. The interested reader will find a more detailed discussion, with a thorough comparison with existing methods, in [6, 7].

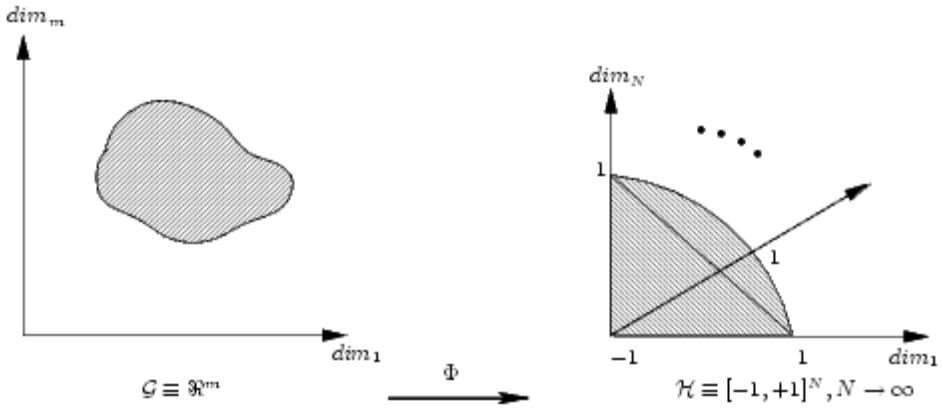

Figure 3: The kernel trick maps the data from a lower dimension space $\mathcal{G} \equiv \Re^m$ to a higher dimension space $\mathcal{H} \equiv [-1, +1]^N, N \to \infty$. This permits to use the Hopfield energy in a MRF framework.

4.1 Spin Glass-Markov Random Fields: Model Derivation

Consider the following energy function:

$$E = -\sum_{(i,j)=1}^{N} J_{ij} s_i s_j, \tag{10}$$

where the s_i are random variables taking values in $[-1, +1]^N$, $\boldsymbol{s} = (s_1, \ldots, s_N)$ is a configuration and $\boldsymbol{J} = [J_{ij}], (i, j) = 1, \ldots, N$ is the connection matrix, $J_{ij} \in [\pm 1]$. Eq (10) is the most general Spin Glass (SG) energy function [1]; the study of the properties of this energy for different $\boldsymbol{J}$s has been a lively area of research in the statistical physics community for the last 25 years.

An important branch in the research area of statistical physics of SG is represented by the application of this knowledge for modeling brain functions. The simplest and most famous SG model of an associative memory was proposed by Hopfield; it assumes J_{ij} to be given by

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^{(\mu)} \, \xi_j^{(\mu)} \, , \tag{11}$$

where the p sets of $\{\boldsymbol{\xi}^{(\mu)}\}_{\mu=1}^{p}, \boldsymbol{\xi}^{(\mu)} \in [-1,+1]^N$ are given configurations of the system (that we call *prototypes*) having the following properties:

$$(a) \quad \boldsymbol{\xi}^{(\mu)} \perp \boldsymbol{\xi}^{(\nu)}, \ \forall \mu \neq \nu; \quad (aa) \quad p << N, \ N \to \infty.$$

Under these assumptions it has been proved that the $\{\boldsymbol{\xi}^{(\mu)}\}_{\mu=1}^{p}$ are the absolute minima of the energy (10); for $\alpha > 0.14$ the system loses its storage capability [1]. These results can be extended from the discrete to the continuous case (i.e $\boldsymbol{s} \in [-1,+1]^N$, see [1]); note that this extension is crucial in the construction of the SG-MRF model.

It is interesting to note that the energy (10), with the prescription (11), can be written as:

$$E = -\frac{1}{N} \sum_{(i,j)=1}^{N} \sum_{\mu=1}^{p} \xi_i^{(\mu)} \xi_j^{(\mu)} s_i s_j = -\frac{1}{N} \sum_{\mu=1}^{p} \sum_{i=1}^{N} (\xi_i^{(\mu)} s_i) \sum_j (\xi_j^{(\mu)} s_j) = -\frac{1}{N} \sum_{\mu=1}^{p} (\boldsymbol{\xi}^{(\mu)} \cdot \boldsymbol{s})^2. \tag{12}$$

Eq (12) depends on the data through scalar products, thus it can be *kernelized*, as to say it can be written as

$$E_{KAM} = -\frac{1}{N} \sum_{\mu=1}^{p} [K(\boldsymbol{\xi}^{(\mu)} \cdot \boldsymbol{s})]^2. \tag{13}$$

The idea to substitute a kernel function, representing the scalar product in a higher dimensional space, in algorithms depending on just the scalar products between data is the so called *kernel trick* [17], which was first used for Support Vector Machines (SVM); in the last few years theoretical and experimental results have increased the interest within the machine learning and computer vision community regarding the use of kernel functions in methods for classification, regression, clustering, density estimation and so on. We call the energy given by eq (13) Kernel Associative Memory (KAM). We can look at eq (12) as follows:

$$E = -\frac{1}{N} \sum_{\mu=1}^{p} (\boldsymbol{\xi}^{(\mu)} \cdot \boldsymbol{s})^2 = -\frac{1}{N} \sum_{\mu=1}^{p} [\Phi(\boldsymbol{\xi}^{(\mu)}) \cdot \Phi(\boldsymbol{s})]^2 = -\frac{1}{N} \sum_{\mu=1}^{p} [K(\boldsymbol{\xi}^{(\mu)} \cdot \boldsymbol{s})]^2 \tag{14}$$

provided that Φ is a mapping such that (see Figure 1):

$$\Phi : \mathcal{G} \equiv \Re^m \mapsto \mathcal{H} \equiv [-1,+1]^N, N \to \infty.$$

that in terms of kernel means

$$K(\boldsymbol{x}, \boldsymbol{x}) = 1, \quad \forall \boldsymbol{x} \in \mathcal{G}, \dim(\mathcal{H}) = N, \quad N \to \infty. \tag{15}$$

If we can find such a kernel, then we can use the KAM energy, with all its properties, for MRF modeling. As the energy is fully connected and the minima of the energy are built by construction, using this energy overcomes all the modeling problems relative to irregular sites for MRF [13]. Conditions (15) are satisfied by generalized Gaussian kernels (9). Regarding the choice of prototypes, given a set of n_k training examples $\{\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \ldots, \boldsymbol{x}_{n_k}^k\}$ for the object class Ω_k, the condition to be satisfied by the p_k prototypes of pattern class k is $\boldsymbol{\xi}^{(\mu)} \perp \boldsymbol{\xi}^{(\nu)} \quad \forall \mu \neq \nu, \quad \mu = 1, \ldots, p_k, \quad p_k << N$ in the mapped space $\mathcal{H}$, that becomes $\Phi(\tilde{\boldsymbol{x}}^{(\mu)}) \perp \Phi(\tilde{\boldsymbol{x}}^{(\nu)}), \quad \forall \mu \neq \nu, \quad \mu = 1, \ldots, p_k, \quad p_k << \dim(\mathcal{H})$ in the data space $\mathcal{G}$.

The measure of the orthogonality of the mapped patterns is the kernel function (9) that, due to the particular properties of Gaussian kernels, has the effect of orthogonalizing the patterns in the space H (see Figure 2). Thus, the orthogonality condition is satisfied by default: if we do not want to introduce further criteria for the choice of prototypes, the natural conclusion is to take all the training samples as prototypes. This approximation is called the *naive ansatz*.

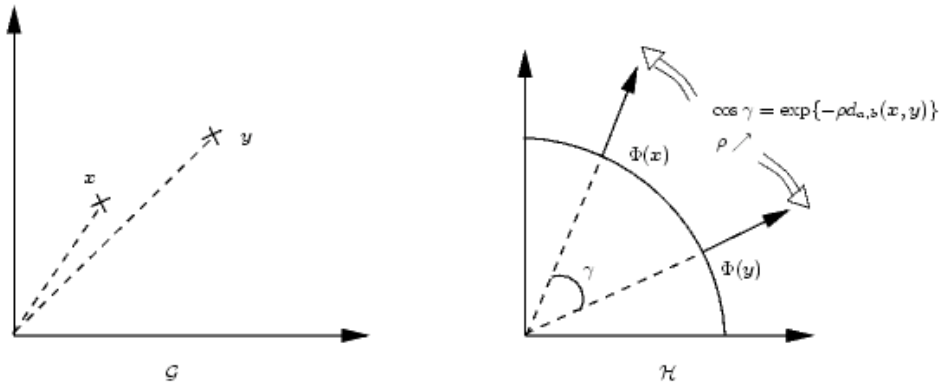

Figure 4: Generalized Gaussian kernels map the data to an infinite dimension hyper-sphere of radius unity. Thus, with a proper choice of ρ, it is possible to orthogonalize all the training data in that space.

5 Kernel Class Specific Classifier

In this section we show how the combination of CSC and SG-MRF leads to a new kernel classifier which fully uses the power of both ideas. First of all, recall that a *kernel* is a function K such that, for all $\boldsymbol{x}, \boldsymbol{y} \in X$,

$$K(\boldsymbol{x}, \boldsymbol{y}) = \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{y}),$$

where Φ is a mapping from X to an (inner product) feature space F [17]. The notation $(\boldsymbol{x} \cdot \boldsymbol{y})$ indicates the scalar product. The SG-MRF energy function can be rewritten as:

$$E_{SG-MRF} = -\sum_{\mu=1}^{p_k} [K(\boldsymbol{x}, \tilde{\boldsymbol{x}}^\mu)]^2 = -\sum_{\mu=1}^{p_k} \tilde{K}(\boldsymbol{x}, \tilde{\boldsymbol{x}}^\mu),$$

where $\tilde{K} = [K(\boldsymbol{x}, \tilde{\boldsymbol{x}}^\mu)]^2$ represents a new kernel function [17]. So we can write the SG-MRF energy as

$$E_{SG-MRF} = -\sum_{\mu=1}^{p_k} \tilde{K}(\boldsymbol{x}, \tilde{\boldsymbol{x}}^\mu) = -\sum_{\mu=1}^{p_k} \Phi(\boldsymbol{x}) \cdot \Phi(\tilde{\boldsymbol{x}}^\mu),$$

and the SG-MRF probability distribution becomes

$$P_{SG-MRF}(\boldsymbol{x}|H_k) = \frac{1}{Z} \exp\left[\sum_{\mu=1}^{p_k} \Phi(\boldsymbol{x}) \cdot \Phi(\tilde{\boldsymbol{x}}^\mu)\right], \tag{16}$$

with Z partition function [13]. Eq (16) tells that P_{SG-MRF} depends on $\boldsymbol{x}$ via a mapping $\Phi(\boldsymbol{x}) = \boldsymbol{z}$. Thus, we can use this probability in the CSC classifier (7), identifying the feature extraction operator $T_k(\boldsymbol{x})$ with the mapping $\Phi_k(\boldsymbol{x})$, as to say using a different mapping, and thus a *different kernel* for each class. We get:

$$k^* = \underset{k=1,\ldots\mathcal{K}}{\operatorname{argmax}} \frac{P_{SG-MRF}(\Phi_k(\boldsymbol{x})|H_k)}{P_{SG-MRF}(\Phi_k(\boldsymbol{x})|H_0)}$$

$$= \underset{k=1,\ldots\mathcal{K}}{\operatorname{argmax}} \left\{ \frac{\frac{1}{Z} \exp\left[\sum_{\mu_k=1}^{p_k} \Phi_k(\boldsymbol{x}) \cdot \Phi_k(\tilde{\boldsymbol{x}}^{\mu_k})\right]}{\frac{1}{Z} \exp\left[\sum_{\mu_0=1}^{p_0} \Phi_k(\boldsymbol{x}) \cdot \Phi_k(\tilde{\boldsymbol{x}}^{\mu_0})\right]} \right\}$$

$$= \underset{k=1,\dots\mathcal{K}}{\operatorname{argmax}} \left\{ \frac{\frac{1}{\exp\left(\frac{1}{N_k}\right)p_k} \exp\left[\sum_{\mu_k=1}^{p_k} \Phi_k(\boldsymbol{x}) \cdot \Phi_k(\tilde{\boldsymbol{x}}^{\mu_k})\right]}{\frac{1}{\exp\left(\frac{1}{N_0}\right)p_0} \exp\left[\sum_{\mu_0=1}^{p_0} \Phi_k(\boldsymbol{x}) \cdot \Phi_k(\tilde{\boldsymbol{x}}^{\mu_0})\right]} \right\}$$

where $\{\tilde{\boldsymbol{x}}^{\mu_k}\}, \mu_k = 1, \dots p_k$ are the set of prototypes of class H_k; $\{\tilde{\boldsymbol{x}}^{\mu_0}\}, \mu_0 = 1, \dots p_0$ are the set of prototypes of class H_0. N_k is the dimension of the space where the PDF representing class H_k is mapped by the mapping Φ_k; N_0 is the dimension of the space where the PDF representing the reference hypothesis H_0 is mapped by the mapping Φ_k. Note that, as the mapping Φ_k is the same for the numerator and denominator, $N_k = N_0, \forall k = 1, \dots \mathcal{K}$. Thus, it follows:

$$= \underset{k=1,\dots\mathcal{K}}{\operatorname{argmax}} \left\{ \frac{p_k}{p_0} \exp\left[\sum_{\mu_k=1}^{p_k} \Phi_k(\boldsymbol{x}) \cdot \Phi_k(\tilde{\boldsymbol{x}}^{\mu_k}) - \sum_{\mu_0=1}^{p_0} \Phi_k(\boldsymbol{x}) \cdot \Phi_k(\tilde{\boldsymbol{x}}^{\mu_0}) \right] \right\}$$

$$= \underset{k=1,\dots\mathcal{K}}{\operatorname{argmin}} \left\{ \ln \frac{p_k}{p_0} \left[- \sum_{\mu_k=1}^{p_k} \Phi_k(\boldsymbol{x}) \cdot \Phi_k(\tilde{\boldsymbol{x}}^{\mu_k}) + \sum_{\mu_0=1}^{p_0} \Phi_k(\boldsymbol{x}) \cdot \Phi_k(\tilde{\boldsymbol{x}}^{\mu_0}) \right] \right\}.$$

Thus, the CSC united to SG-MRF gives the *Kernel-Class Specific Classifier*:

$$k^* = \underset{k=1,\dots\mathcal{K}}{\operatorname{argmin}} \left\{ \ln \frac{p_k}{p_0} \left[- \sum_{\mu_k=1}^{p_k} [K_k(\boldsymbol{x}, \tilde{\boldsymbol{x}}^{\mu_k})]^2 + \sum_{\mu_0=1}^{p_0} [K_k(\boldsymbol{x}, \tilde{\boldsymbol{x}}^{\mu_0})]^2 \right] \right\}. \tag{17}$$

If the number of prototypes is the same for all the classes $\{H_k\}_{k=1}^{\mathcal{K}}$ and for the reference hypothesis H_0, then equation (17) becomes

$$k^* = \underset{k=1,\dots\mathcal{K}}{\operatorname{argmin}} \left[- \sum_{\mu_k=1}^{p_k} [K_k(\boldsymbol{x}, \tilde{\boldsymbol{x}}^{\mu_k})]^2 + \sum_{\mu_0=1}^{p_0} [K_k(\boldsymbol{x}, \tilde{\boldsymbol{x}}^{\mu_0})]^2 \right]. \tag{18}$$

Given a training set, the kernel K_k can be *learned*, for each class H_k, as for SG-MRF (section 4). Thus, K-CSC permits to use a different representation for each class, according to its needs, as CSC does. But as the K-CSC representation is bound to be a specific class of kernels, it solves the ambiguity of CSC regarding the choice of the representations and permits to learn them. Moreover, this permits to use a different kernel *and* a different set of features for each class.

The reader could wonder whether the $\boldsymbol{z}_k = \Phi_k(\boldsymbol{x})$ are a sufficient statistics for the class H_k, as required by CSCs. It could be argued that it is not, as the mapping Φ_k is a mapping in a higher dimensional space. The point is that, although Φ_k maps the data into a higher dimensional space, it can be proved that the mapped data are embedded in a subspace of the mapped space F, which will be of dimension lower or equal to the dimension of the data set [17]. Thus, the problem of having a sufficient statistic in the mapped space is the same of having a sufficient statistic in the original feature space. This is in turn related to having enough training data, given the space where one whishes to estimate the PDF. Therefore, it is ultimately related to the dimension of the space where the PDF is estimated. Indeed, the problem of having a sufficient statistic is the problem of having enough training data in order to make a correct estimate of the PDF in the data space with a given dimension. We can therefore conclude that, if $\boldsymbol{x}$ is a sufficient statistic for the class H_k, so it is $\Phi_k(\boldsymbol{x})$.

6 Experiments

The application of CSC to visual recognition is possible in principle, but it has been very challenging until now [2]. Still, SG-MRFs have shown very good performances in the domain of object recognition [7, 6]. Thus, we decided to test K-CSC on this task. We ran two sets of experiments, one to assess the method in case of objects

imaged in increasingly difficult conditions, the other to check the performance of K-CSC with respect to an increasing number of classes. In both cases, the chosen databases contain images of different sizes. Thus, we represented each view with Gaussian derivative histograms [16] using Gaussian derivative filters along x and y, variance $\sigma = 1.0$ and resolution for histogram axis of 16 bins:

$$D_x = -\frac{x}{\sigma^2}G^\sigma(x,y), \qquad D_y = -\frac{y}{\sigma^2}G^\sigma(x,y),$$

where $G^\sigma(x,y)$ is the Gaussian distribution. We chose these features because they have been extensively used with SG-MRFs on several applications, obtaining good results [7, 6]. From the point of view of specific features, this is equivalent to do, for each image, $\boldsymbol{x} \rightarrow D_x D_y(\boldsymbol{x}) \rightarrow \Phi(D_x D_y(\boldsymbol{x}))$. The functional form of the kernel for K-CSC is fixed and given by eq (9). The parameters (a, b, ρ) need to be learned for each class. To do so, we used a gradient descent algorithm.

The K-CSC algorithm has one parameter that must be set by the user: the reference hypothesis H_0. Here we decided to use one of the object classes, selected during the training step. This has the double advantage of limiting the field of possible choices for H_0, and not having to estimate an additional probability density function. Although we cannot claim that this strategy for the choice of H_0 is optimal, we argue that it is reasonable. Indeed, the underlying principle in doing classification using probability ratios is to perform multi--class discrimination against a 'non-object' class. If this class is too broad (a common strategy is to define a 'background' class, see for instance [10]) there are two possible risks:

1. If the estimate of the probability density function is done on a training set which is not representative enough of the reference class, the resulting probability ratio classifier will be biased toward the object class we are interested in, leading to a very high number of false positive.

2. If the class is too broad and it is estimated effectively, it might become a strong attractor for almost any class, and the resulting classifier might be biased toward the reference hypothesis, resulting in a very high number of false negative (this might happen for instance in the case of distinguishing objects like cups, books and so on, if we would choose as reference hypothesis the class 'desk background').

By choosing as reference hypothesis one of the objects we want to classify, we make sure that we will discriminate with respect to well-defined visual classes, and we have a reasonable hope that the training data from which we estimate the probability density functions are a sufficient statistic for the reference class H_0 (if this would not be the case, we would be in trouble anyway for the estimate of the PDF of that object class). Last but not least, this means that, for a N-class recognition problem, we must estimate N probability density functions, and not $N + 1$.

For all experiments, we benchmarked K-CSC against SG-MRF and support vector machines [17] with Gaussian kernel. For both methods, kernel parameters were selected via cross-validation. In the following we describe the databases and the results obtained for each set of experiments (section 6.1 and 6.2).

6.1 First Set of Experiments

We ran a first set of experiments on a database of 6 objects [15]: a cup, a fighter, a plane, a car, a toy rabbit and a toy bear. The database contains, for each object, a training set of 106 views (53 for the car) taken on a sphere (hemisphere for the car) approximately every 20 degrees, in black homogeneous background. There are four different test sets: one taken in black homogeneous background, one in black homogeneous background with distracting objects around, one on a white marble table and one on a poster with Christmas decorations. Figure 5 shows examples of the object 'cup' taken in these four different backgrounds. Each of these sets has 53 (24 for the car) views, positioned in between the training images. The test views are taken at the same scale of those in the training set, but the illumination conditions change from background to background.

In a first series of experiments, we used the training and test set described above. Results are reported in Table 1. We see that for all experiments, K-CSC, using a specific reference hypothesis, gives the best performance.

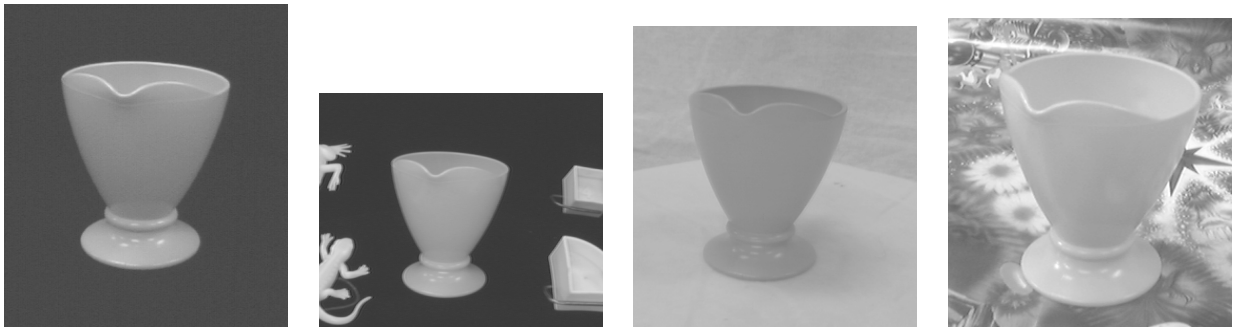

Figure 5: The cup from the 6 object database [15] in (from left to right) homogeneous background, with distracting objects, in white background and in textured background.

	homo	**heter1**	**heter2**	**heter3**
SG-MRF	99.24 %	80.68 %	32.95 %	35.98 %
SVM	99.30 %	80.56 %	37.15 %	40.61 %
K − CSC	**99.65 %**	**87.16 %**	**55.21 %**	**49.30 %**

Table 1: Results for the first series of experiments. Training was done on object views taken in homogeneous background. Test was done on object views taken in four different backgrounds.

In three cases out of four, SVM with Gaussian kernel obtains the second best performance. It should be noted though that in two cases out of four (backgrounds 'homo' and 'heter1') the performance of SG-MRF and SVM are very close, and can be considered equivalent.

Table 1 shows a strong improvement in performance between SG-MRF and K-CSC, for all test sets. This is an important result, because in both cases we are using the same method for evaluating the PDFs. Thus, the better performance depends by the CSC approach. Nevertheless, the performance for images in heterogeneous background is generally poor. This mostly depends on the fact that we are using as starting data $\boldsymbol{x}$ a feature representation (Gaussian derivative histograms) which is global, and it is well known that global representations suffer for changes of background with respect to the training set. But it is also known that SG-MRFs are quite robust to degradation in the test set, provided that a reasonable amount of degradation is introduced as well in the training set (see [7] and references therein)[†]. Thus, we ran a second series of experiments using as training set the old training set plus 1/4 of views taken from all the previous test sets. Views were chosen randomly, and test sets consisted of the remaining views. Results are reported in Table 2, and confirm the previous results as well the robustness properties of SG-MRF. Note that SVM too benefits from this approach.

	homo	**heter1**	**heter2**	**heter3**
SG-MRF	99.24 %	91.29 %	85.98 %	85.98 %
SVM	99.30 %	90.28 %	88.78 %	89.29 %
K-CSC	**99.65 %**	**93.40 %**	**91.29 %**	**91.67 %**

Table 2: Results for the second series of experiments. Training was done on object views taken in homogeneous and heterogeneous background. Test was done on object views taken in four different backgrounds. Training and test set are disjoint.

[†]This is similar in spirit to the idea behind Virtual Support Vector Machines, [8].

6.2 Second Set of Experiments

In the second set of experiments we evaluate the performance of K-CSC, SVM and SG-MRF for an increasing number of classes. We ran experiments on a database of 59 objects [15]: 11 cups, 5 dolls, 6 planes, 6 fighter jets, 9 lizards, 5 spoons, 8 snakes and 9 sport cars. Each object is represented in the training set by a collection of views taken approximately every 20 degrees on a sphere. This amounts to 106 views for a full sphere, and 53 for a hemisphere. The test set consists of 48 (24) views, positioned in between the training images, and taken under the same conditions. Cups, dolls, fighters, planes, spoons are represented by 106 views in the training set and 48 views in the test set. Lizards, snakes, sport cars are represented by 53 views in the training set and 24 views in the test set. We performed 6 experiments, with an increasing number of objects: 1 object for each category (8 classes), 2 objects for each category (16 classes), and so on, until we reached 5 objects for each category (40 classes). The last experiment uses all available objects (59 classes). Results are shown in Table 3. We see that, for all experiments, K-CSC achieves the best performance, followed by SVMs and then SG-MRFs. The difference in performance between SG-MRF and SVM is not very pronounced, as observed in the previous set of experiments. It is interesting to note that, for all three algorithms, there is a decrease in performance as the number of classes grows. Still, this is far less pronounced for K-CSC, showing that the class specific strategy clearly pays off in terms of scalability.

From these results, and the experimental findings of the first set of experiments, we can conclude that K-CSC are an effective method for class specific recognition of visual classes.

	8 classes	**16 classes**	**24 classes**	**32 classes**	**40 classes**	**59 classes**
SG-MRF	99.24 %	94.30 %	90.28 %	88.78 %	85.98%	81.06 %
SVM	99.24 %	95.15 %	93.40 %	91.47 %	89.29 %	85.98%
K-CSC	**99.65 %**	**99.24 %**	**98.86 %**	**97.30 %**	**95.86 %**	**93.41 %**

Table 3: Results for the second set of experiments. Training was done on an incresing number of object classes, so to test scalability. K-CSC obtains the best performance, showing very good robustness properties with respect to the increasing number of classes.

7 Conclusions

In this paper we presented a new kernel classifier that permits us to use different kernels for different classes. We achieve this result combining a Gibbs kernel distribution, SG-MRF, with the feature-based class specific classifier. The novel classifier presents several advantages: first, it permits to learn the optimal kernel to be used for each class. This results in a remarkable increase in the recognition rate with respect to SG-MRF. Second, it proposes a novel approach to the problem of optimal features selection for a specific classification problem. Finally, it makes it straightforward to use the class specific classifier approach to visual recognition, a challenging task until now.

This work can be extended in many ways. To begin with, we would like to test the method on the object categorization problem, and more generally on other visual recognition problems like texture classification, action recognition and so on. Here we benchmarked our aproach with respect to learning methods, like SVM and nearest neighbors. An equally important comparison would be with state of the art feature selection methods. An open issue is the very expensive learning procedure, which might limit the use of the method for real-world applications with a large number of classes. A possible solution might be to use a faster classifier (as SG-MRF) in a first step to generate hypotheses, and then use K-CSC in a second step, on the subset of the best n hypotheses. This would permit to control the number of classes for the K-CSC classifier. The number of hypotheses might be determined in a principled way, considering the recognition time and the number of false positive, as suggested by Viola and Jones in [21]. Future work will explore these points.

References

[1] D. J. Amit, *Modeling Brain Function*, Cambridge University Press, Cambridge, USA, 1989.

[2] P. M. Baggenstoss, H. Niemann, "A theoretically optimal probabilistic classifier using class-specific features", *IEEE Proc of Intl. Conf on Pattern Recognition*, pp 2763-2768, 2000.

[3] P. M. Baggenstoss, "Class-specific features in classification", *IEEE Transaction on Signal Processing*, 47, 3428 -3432, 1999.

[4] S. Belongie, J. Malik, J. Puchiza, "Shape matching and object recognition using shape contexts", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(4): 509-522, 2002.

[5] C. M. Bishop, *Neural Networks for Pattern Recognition*, Claredon Press, Oxford, 1995.

[6] B. Caputo, "Spin glass models of Markov random fields", *International Journal on Image System and Technology*, 16(5): 181-188, 2007.

[7] B. Caputo. *A new kernel method for object recognition: spin glass-Markov random fields*. PhD Thesis, Royal Institute of Technology, November 2004. Available at http://people.idiap.ch/caputo.

[8] D. Decoste, B. Schoelkopf, "Training invariant support vector machines", *Machine Learning*, 46: 161-190, 2002.

[9] R. O. Duda, P. E. Hart, D. G. Stork. *Pattern Classification*. Wiley Interscience, 2000.

[10] R. Fergus, P. Perona, A. Zissermann. "Object class recognition by unsupervised scale-invariant learning", *IEEE Proc of Intl. Conf. on Computer Vision and Pattern Recognition*, 2:264-271, 2003.

[11] D. Koller, M. Sahami, "Toward optimal feature selection", *Proc of the Thirtheenth International Conference on Machine Learning*, 284-292,1996.

[12] B. Leibe, B. Schiele, "Analyzing appearance and contour based methods for object categorization", *IEEE Proc of Intl. Conf. on Computer Vision and Pattern Recognition*, 2:409-415, 2003.

[13] S. Z. Li, *"Markov Random Field Modeling in Computer Vision"*, Springer-Verlag, 1995.

[14] J. Matas, R. Marik, J. Kittler, "On representation and matching of multi-coloured objects", *IEEE Proc of the Intl. Conf. on Computer Vision*, 726-732, 1995.

[15] R. Nelson and A. Selinger, "A cubist approach to object recognition", *IEEE Proc of Intl. Conf. of Computer Vision*, 614-621, 1998.

[16] B. Schiele, J. L. Crowley, "Recognition without correspondence using multidimensional receptive field histograms", *International Journal on Computer Vision*, 36 (1): 31- 52, 2000.

[17] B. Schölkopf, A. J. Smola, *Learning with kernels*, 2002, the MIT Press.

[18] C. Schmid, R. Mohr, C. Bauckhage, "Evaluation of Interest Point Detectors", *International Journal of Computer Vision*, 37(2): 151-172, 2000.

[19] H. Schneiderman , T. Kanade, "A statistical method of 3d object detection applied to faces and cars", *IEEE Proc of the Intl. Conf. on Computer Vision and Pattern Recognition*, 1746-1759, 2000.

[20] M. J. Swain, D. H. Ballard, "Color Indexing", *Interational Journal of Computer Vision*, 7(1): 11-32, 1991.

[21] P. Viola, M. Jones, "Robust real-time object detection", *International Journal on Computer Vision*, 57(2): 137-154, 2004.

[22] S. C. Zhu, Y. N. Wu and D.B. Mumford, "FRAME: Filters, Random field And Maximum Entropy: Towards a Unified Theory for Texture Modeling", *International Journal of Computer Vision*, 27(2): 1-20, 1998.