Recognizing 3-D Objects with Linear Support Vector Machines

Massimiliano Pontil[1], Stefano Rogai[2], and Alessandro Verri[3]

[1] Center for Biological and Computational Learning, MIT, Cambridge MA (USA)
[2] INFM - DISI, Università di Genova, Genova (I)

Abstract. In this paper we propose a method for 3-D object recognition based on linear Support Vector Machines (SVMs). Intuitively, given a set of points which belong to either of two classes, a linear SVM finds the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. The hyperplane is determined by a subset of the points of the two classes, named *support vectors*, and has a number of interesting theoretical properties. The proposed method does not require feature extraction and performs recognition on images regarded as points of a space of high dimension. We illustrate the potential of the recognition system on a database of 7200 images of 100 different objects. The remarkable recognition rates achieved in all the performed experiments indicate that SVMs are well-suited for aspect-based recognition, even in the presence of small amount of occlusions.

1 Introduction

Support Vector Machines (SVMs) have recently been proposed as a very effective method for general purpose pattern recognition [12, 3]. Intuitively, given a set of points which belong to either of two classes, a SVM finds the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. According to [12], given fixed but unknown probability distributions, this hyperplane – called the Optimal Separating Hyperplane (OSH) – minimizes the risk of misclassifying the *yet-to-be-seen* examples of the test set.

In this paper an aspect-based method for the recognition of 3–D objects which makes use of SVMs is described. In the last few years, aspect-based recognition strategies have received increasing attention from both the psychophysical [10, 4] and computer vision [7, 2, 5] communities. Although not naturally tolerant to occlusions, aspect-based recognition strategies appear to be well-suited for the solution of recognition problems in which geometric models of the viewed objects can be difficult, if not impossible, to obtain. Unlike other aspect-based methods, recognition with SVMs (*a*) does not require feature extraction or data reduction, and (*b*) can be performed directly on images regarded as points of an N-dimensional object space, *without* estimating pose. The high dimensionality of the object space makes OSHs very effective decision surfaces, while the recognition stage is reduced to deciding on which side of an OSH lies a given point in object space.

The proposed method has been tested on the COIL database[3] consisting of 7200 images of 100 objects. Half of the images were used as training examples, the remaining half as test images. We discarded color information and tested the method on the remaining images corrupted by synthetically generated noise, bias, and small amount of occlusions. The remarkable recognition rates achieved in all the performed experiments indicate that SVMs are well-suited for aspect-based recognition. Comparisons with other pattern recognition methods, like perceptrons, show that the proposed method is far more robust in the presence of noise.

The paper is organized as follows. In Section 2 we review the basic facts of the theory of SVMs. Section 3 discusses the implementation of SVMs adopted throughout this paper and describes the main features of the proposed recognition system. The obtained experimental results are illustrated in Section 4. Finally, Section 5 summarizes the conclusions that can be drawn from the presented research.

2 Theoretical overview

We recall here the basic notions of the theory of SVMs [12, 3]. We start with the simple case of linearly separable sets. Then we define the concept of support vectors and deal with the more general nonseparable case. Finally, we list the main properties of SVMs. Since we have only used linear SVMs we do not cover the generalization of the theory to the case of nonlinear separating surfaces.

2.1 Optimal separating hyperplane

In what follows we assume we are given a set S of points $\mathbf{x}_i \in \mathbb{R}^n$ with $i = 1, 2, \ldots, N$. Each point $\mathbf{x}_i$ belongs to either of two classes and thus is given a label $y_i \in \{-1, 1\}$. The goal is to establish the equation of a hyperplane that divides S leaving all the points of the same class on the same side while maximizing the distance between the two classes and the hyperplane. To this purpose we need some preliminary definitions.

Definition 1. The set S is *linearly separable* if there exist $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \tag{1}$$

for $i = 1, 2, \ldots, N$.

The pair $(\mathbf{w}, b)$ defines a hyperplane of equation $\mathbf{w} \cdot \mathbf{x} + b = 0$ named the *separating hyperplane* (see Figure 1(a)). If we denote by w the norm of $\mathbf{w}$, the

[3] The images of the COIL database (Columbia Object Image Library) can be downloaded through anonymous ftp from www.cs.columbia.edu.

signed distance d_i of a point $\mathbf{x}_i$ from the separating hyperplane $(\mathbf{w}, b)$ is given by

$$d_i = \frac{\mathbf{w} \cdot \mathbf{x}_i + b}{w}, \tag{2}$$

with w the norm of $\mathbf{w}$. Combining inequality (1) and equation (2), for all $x_i \in S$ we have

$$y_i d_i \geq \frac{1}{w}. \tag{3}$$

Therefore, $1/w$ is the lower bound on the distance between the points $\mathbf{x}_i$ and the separating hyperplane $(\mathbf{w}, b)$.

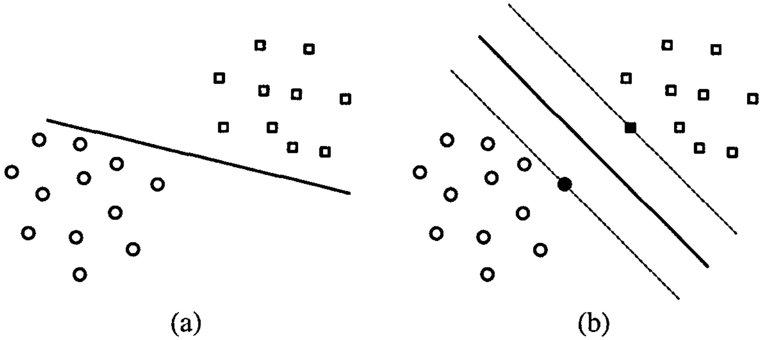

Fig. 1. Separating hyperplane (a) and OSH (b). The dashed lines in (b) identify the margin.

We now need to establish a one-to-one correspondence between separating hyperplanes and their parametric representation.

Definition 2. Given a separating hyperplane $(\mathbf{w}, b)$ for the linearly separable set S, the *canonical representation* of the separating hyperplane is obtained by rescaling the pair $(\mathbf{w}, b)$ into the pair $(\mathbf{w}', b')$ in such a way that the distance of the closest point, say $\mathbf{x}_j$, equals $1/w'$.

Through this definition we have

$$\min_{\mathbf{x}_i \in S} \{y_i(\mathbf{w}' \cdot \mathbf{x}_i + b')\} = 1.$$

Consequently, for a separating hyperplane in the canonical representation, the bound in inequality (3) is tight. In what follows we will assume that a separating hyperplane is always given in the canonical representation and thus write $(\mathbf{w}, b)$ instead of $(\mathbf{w}', b')$. We are now in a position to define the notion of OSH.

Definition 3. Given a linearly separable set S, the *optimal separating hyperplane* is the separating hyperplane for which the distance of the closest point of S is maximum.

Since the distance of the closest point equals $1/w$, the OSH can be regarded as the solution of the problem of minimizing $1/w$ subject to the constraint (1), or

Problem **P1**
Minimize $\frac{1}{2}\mathbf{w}\cdot\mathbf{w}$
subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1,\ i=1,2,\ldots,N$

Note that the parameter b enters in the constraints but not in the function to be minimized. The quantity $2/w$, the lower bound of the minimum distance between points of different classes, is named the *margin*. Hence, the OSH can also be seen as the separating hyperplane which maximizes the margin (see Figure 1(b)). We now study the properties of the solution of Problem **P1**.

2.2 Support vectors

Problem **P1** is usually solved by means of the classical method of Lagrange multipliers. In order to understand the concept of SVs it is necessary to go briefly through this method. For more details and a thorough review of the method see [1].

If we denote with $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\ldots,\alpha_N)$ the N nonnegative Lagrange multipliers associated with the constraints (1), the solution to Problem **P1** is equivalent to determining the *saddle point* of the function

$$L=\frac{1}{2}\mathbf{w}\cdot\mathbf{w}-\sum_{i=1}^{N}\alpha_i\left\{y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1\right\}. \tag{4}$$

with L = L(**w**, b, $\boldsymbol{\alpha}$). At the saddle point, L has a minimum for $\mathbf{w}=\bar{\mathbf{w}}$ and $b=\bar{b}$ and a maximum for $\boldsymbol{\alpha}=\bar{\boldsymbol{\alpha}}$, and thus we can write

$$\frac{\partial L}{\partial b}=\sum_{i=1}^{N}y_i\alpha_i=0, \tag{5}$$

$$\frac{\partial L}{\partial \mathbf{w}}=\mathbf{w}-\sum_{i=1}^{N}\alpha_i y_i\mathbf{x}_i=0 \tag{6}$$

with

$$\frac{\partial L}{\partial \mathbf{w}}=(\frac{\partial L}{\partial w_1},\frac{\partial L}{\partial w_2},\ldots,\frac{\partial L}{\partial w_N}).$$

Substituting equations (5) and (6) into the right hand side of (4), we see that Problem **P1** reduces to the maximization of the function

$$\mathcal{L}(\boldsymbol{\alpha})=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j\mathbf{x}_i\cdot\mathbf{x}_j, \tag{7}$$

subject to the constraint (5) with $\boldsymbol{\alpha}\geq 0$[4]. This new problem is called the *dual problem* and can be formulated as

[4] In what follows $\boldsymbol{\alpha}\geq 0$ means $\alpha_i\geq 0$ for every component α_i of any vector $\boldsymbol{\alpha}$.

Problem **P2**
Maximize $-\frac{1}{2}\alpha^{\top} D\alpha + \sum \alpha_i$
subject to $\sum y_i \alpha_i = 0$
$\alpha \geq 0,$

where both sums are for $i = 1, 2, \ldots, N$, and D is an $N \times N$ matrix such that

$$D_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j. \tag{8}$$

As for the pair $(\bar{\mathbf{w}}, \bar{b})$, from equation (6) it follows that

$$\bar{\mathbf{w}} = \sum_{i=1}^{N} \bar{\alpha}_i y_i \mathbf{x}_i, \tag{9}$$

while $\bar{b}$ can be determined from $\bar{\alpha}$, solution of the dual problem, and from the Kühn-Tucker conditions

$$\bar{\alpha}_i \left(y_i (\bar{\mathbf{w}} \cdot \mathbf{x}_i + \bar{b}) - 1 \right) = 0, \quad i = 1, 2, \ldots, N. \tag{10}$$

Note that the only $\bar{\alpha}_i$ that can be nonzero in equation (10) are those for which the constraints (1) are satisfied with the equality sign. This has an important consequence. Since most of the $\bar{\alpha}_i$ are usually null, the vector $\bar{\mathbf{w}}$ is a linear combination of a relatively small percentage of the points $\mathbf{x}_i$. These points are termed *support vectors* (SVs) because they are the closest points from the OSH and the only points of S needed to determine the OSH (see Figure 1(b)). Given a support vector $\mathbf{x}_j$, the parameter $\bar{b}$ can be obtained from the corresponding Kühn-Tucker condition as $\bar{b} = y_j - \bar{\mathbf{w}} \cdot \mathbf{x}_j$.

2.3 Linearly nonseparable case

If the set S is not linearly separable or one simply ignores whether or not the set S is linearly separable, the problem of searching for an OSH is meaningless (there may be no separating hyperplane to start with). Fortunately, the previous analysis can be generalized by introducing N nonnegative variables $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_N)$ such that

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N. \tag{11}$$

The purpose of the variables ξ_i is to allow for a small number of misclassified points. If the point $\mathbf{x}_i$ satisfies inequality (1), then ξ_i is null and (11) reduces to (1). Instead, if the point $\mathbf{x}_i$ does not satisfy inequality (1), the extraterm $-\xi_i$ is added to the right hand side of (1) to obtain inequality (11). The generalized OSH is then regarded as the solution to

Problem **P3**
Minimize $\frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum \xi_i$
subject to $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \; i = 1, 2, \ldots, N$
$\boldsymbol{\xi} \geq 0.$

The purpose of the extraterm $C\sum\xi_i$, where the sum is for $i = 1, 2, \ldots, N$, is to keep under control the number of misclassified points. The parameter C can be regarded as a regularization parameter. The OSH tends to maximize the minimum distance $1/w$ for small C, and minimize the number of misclassified points for large C. For intermediate values of C the solution of problem **P3** trades errors for a larger margin. The behavior of the OSH as a function of C is studied in detail in [8].

In analogy with what was done for the separable case, Problem **P3** can be transformed into the *dual*

$$
\begin{array}{lll}
\text{Problem } \mathbf{P4} & & \\
\text{Maximize} & -\frac{1}{2}\alpha^{\top} D\alpha + \sum \alpha_i & \\
\text{subject to} & \sum y_i\alpha_i = 0 & \\
 & 0 \leq \alpha_i \leq C, & i = 1, 2, \ldots, N
\end{array}
$$

Note that the dimension of **P4** is given by the size of the training set, while the dimension of the input space gives the rank of D. From the constraints of Problem **P4** it follows that if C is sufficiently large and the set S linearly separable, Problem **P4** reduces to **P2**. The vector $\mathbf{w}$ is still given by equation 9, while $\bar{b}$ can again be determined from $\bar{\alpha}$, solution of the dual problem **P4**, and from the new Kuhn-Tucker conditions

$$\bar{\alpha}_i\left(y_i(\bar{\mathbf{w}}\cdot\mathbf{x}_i + \bar{b}) - 1 + \bar{\xi}_i\right) = 0 \qquad (12)$$

$$(C - \bar{\alpha}_i)\bar{\xi}_i = 0 \qquad (13)$$

where the $\bar{\xi}_i$ are the values of the ξ_i at the saddle point. Similarly to the separable case, the SVs are the points $\mathbf{x}_i$ for which $\bar{\alpha}_i > 0$. The main difference is that here we have to distinguish between the SVs for which $\bar{\alpha}_i < C$ and those for which $\bar{\alpha}_i = C$. In the first case, from condition (13) it follows that $\bar{\xi}_i = 0$, and hence, from condition (12), that the SVs lie at a distance $1/\bar{w}$ from the OSH. These SVs are termed *margin vectors.* The SVs for which $\bar{\alpha}_i = C$, are instead: misclassified points if $\xi_i > 1$, points correctly classified but closer than $1/\bar{w}$ from the OSH if $0 < \xi \leq 1$, or margin vectors if $\xi_i = 0$. Neglecting this last rare (and degenerate) occurrence, we refer to all the SVs for which $\alpha_i = C$ as *errors.* All the points that are not SVs are correctly classified and lie outside the margin strip.

We conclude this section by listing the main properties of SVMs.

2.4 Mathematical properties

The first property distinguishes SVMs from previous nonparametric techniques, like nearest-neighbors or neural networks. Typical pattern recognition methods are based on the minimization of the *empirical risk*, that is on the attempt to minimize the misclassification errors on the training set. Instead, SVMs minimize the *structural risk*, that is the probability of misclassifying a previously unseen data point drawn randomly from a fixed but unknown probability distribution. In particular, it follows that, if the VC-dimension [11] of the family of decision surfaces is known, then the theory of SVMs provides an upper bound for the probability of misclassification of the test set for any possible probability distributions of the data points [12].

Secondly, SVMs condense all the information contained in the training set relevant to classification in the support vectors. This (*a*) reduces the size of the training set identifying the most important points, and (*b*) makes it possible to perform classification efficiently.

Thirdly, SVMs can be used to perform classification in high dimensional spaces, even in the presence of a relatively small number of data points. This is because, unlike other techniques, SVMs look for the optimal separating hyperplane. From the quantitative viewpoint, the margin can be used as a measure of the difficulty of the problem (the larger the margin the lower the probability of misclassifying a yet-to-be-seen point).

3 The recognition system

We now describe the recognition system we devised to assess the potential of the theory. We first review the implementation developed for determining the SVs and the associated OSH.

3.1 Implementation

In Section 2 we have seen that the problem of determining the OSH reduces to Problem **P4**, a typical problem of quadratic programming. The vast literature of nonlinear programming covers a multitude of problems of quadratic programming and provides a plethora of methods for their solution. Our implementation makes use of the equivalence between quadratic programming problems and *Linear Complementary* Problems (LCPs) and is based on the *Complementary Pivoting Algorithm* (CPA), a classical algorithm able to solve LCPs [1].

Since CPA spatial complexity goes with the square of the number of examples, the algorithm cannot deal efficiently with much more than a few hundreds of examples. This has not been a fundamental issue for the research described in this paper, but for problems of larger size one definitely has to resort to more sophisticated techniques [6].

3.2 Recognition stages

We have developed a recognition system based on three stages:

1. Preprocessing
2. Training set formation
3. System testing

We now describe these three stages in some detail.

Preprocessing The COIL database consists of 72 images of 100 objects (for a total of 7200 images), objects positioned in the center of a turntable and observed from a fixed viewpoint. For each object, the turntable is rotated by 5° per image. Figures 2 shows a selection of the objects in the database. Figures 3 shows one every three views of one particular object. As explained in detail by Murase and Nayar[5], the object region is re-sampled so that the larger of the two dimensions fits the image size. Consequently, the apparent size of an object may change considerably from image to image, especially for the objects which are not symmetric with respect to the turntable axis.

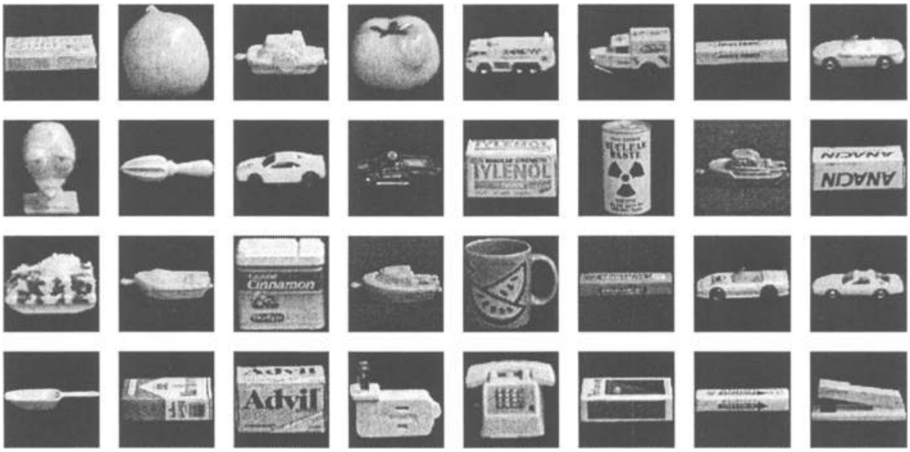

Fig. 2. Images of 32 objects of the COIL database.

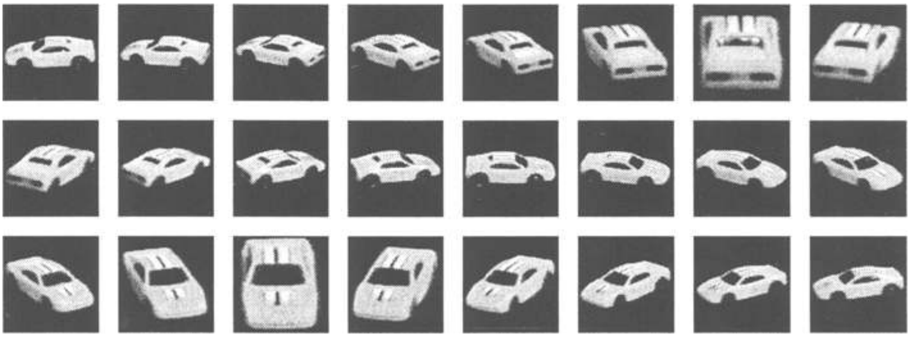

Fig. 3. Twentyfour of the 72 images of a COIL object.

The original images were color images (24 bits for each of the RGB channels) of 128 × 128 pixels. In the preprocessing stage each image was transformed into an 8–bit grey-level image rescaling the obtained range between 0 and 255. Finally, the image spatial resolution was reduced to 32 × 32 by averaging the grey values over 4 × 4 pixel patches. The aim of these transformations was to reduce the dimensionality of the representation given the relatively small number of images available. The effectiveness of this considerable data reduction is explained elsewhere [9].

Forming the training set The training set consists of 36 images (one every 10°) for each object. After the preprocessing stage, each image can be regarded as a vector $\mathbf{x}$ of $32 \times 32 = 1024$ components.

Depending on the classification task, a certain subset of the 100 objects (from 2 to 32) has been considered. Then, the OSHs associated to each pair of objects i and j in the subset were computed, the SVs identified, and the obtained parameters, $\mathbf{w}(i,j)$ and $b(i,j)$, stored in a file. We have never come across errors in the classification of the training sets. The reason is essentially given by the high dimensionality of the object space compared to the small number of examples. The images corresponding to some of the SVs for a specific pair of objects are shown in Figure 4.

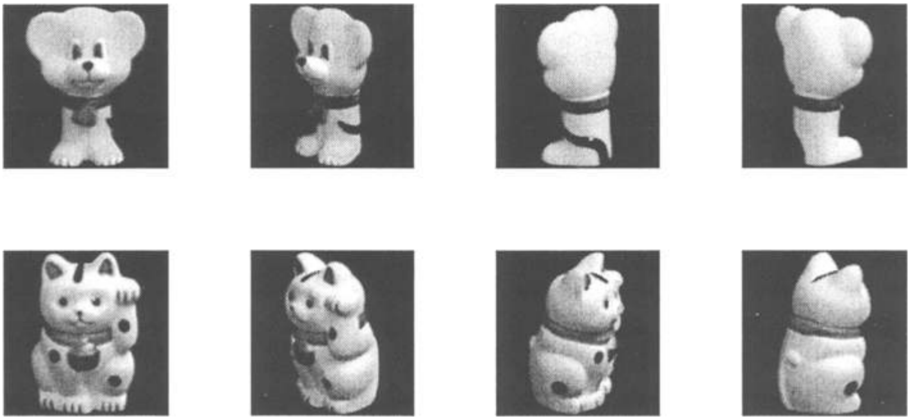

Fig. 4. Eight of the SVs for a specific object pair.

Typically, we have found a number of SVs ranging from 1/3 to 2/3 of the 72 training images for each object pair. This large fraction of SVs can be explained by the high dimensionality of the object space combined with the small number of examples.

System testing Given a certain subset σ of the 100 objects and the associated training set of 36 images for each object in σ, the test set consists of the remaining

36 images per object in σ. Recognition was performed following the rules of a tennis tournament. Each object is regarded as a *player*, and in each *match* the system temporarily classifies an image of the test set according to the OSH relative to the pair of players involved in the match. If in a certain match the players are objects i and j, the system classifies the viewed object of image $\mathbf{x}$ as object i or j depending on the sign of

$$\mathbf{w}(i,j) \cdot \mathbf{x} + b(i,j).$$

If, for simplicity, we assume there are 2^K players, the first round 2^{K-1} matches are played and the 2^{K-1} losing players are out. The 2^{K-1} match winners advance to the second round. The $(K-1)$-th round is the final between the only 2 players that won all the previous matches. This procedure requires $2^K - 1$ classifications. Note that the system recognizes object identity without estimating pose.

We are now ready to present the experimental results.

4 Experimental results

We describe here the experimental results of the recognition system on the COIL database. We first considered the images exactly as downloaded from the Net and afterwords verified what amount of noise the system can tolerate.

4.1 COIL images

We tested the proposed recognition system on sets of 32 of the 100 COIL objects. The training sets consisted of 36 images for each of 32 objects and the test sets the remaining 36 images for each object. For all 10 random choices of 32 of the 100 objects we tried, the system reached perfect score. Therefore, we decided to select by hand the 32 objects *most difficult* to recognize (*i.e.* the set of objects separated by the smallest margins). By doing so the system finally mistook a packet of chewing gum for another very similar packet of chewing gum in one case (see Figure 5).

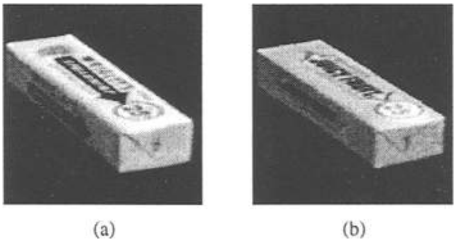

Fig. 5. The only misclassified image (a) and corresponding erroneously recognized object (b).

To gain a better understanding of how an SVM perform recognition, it may be useful to look at the relative weights of the components of the vector $\mathbf{w}$. A grey valued encoded representation of the absolute value of the components of the vector $\mathbf{w}$ relative to the OSH of the two objects of Figure 4 is displayed in Figure 6(a) (the darker a point, the higher the corresponding $\mathbf{w}$ component). Note that the background is essentially irrelevant, while the larger components (in absolute value) can be found in the central portion of the *image*. Interestingly, the image of Figure 6(a) resembles the visual appearance of both the "dog" and "cat" of Figure 4. The graph of Figure 6(b) shows the convergence of $\sum w_i x_i$ to the dot product $\mathbf{w} \cdot \mathbf{x}$ for one of the "cat" image, with the components w_i sorted in decreasing order. From the graph it clearly follows that less than half of the 1024 components are all that is needed to reach almost perfect convergence, while a reasonably good approximation is already obtained using only the largest 100 components. The graph of Figure 6(b) is typical with a few exceptions corresponding to very similar object pairs.

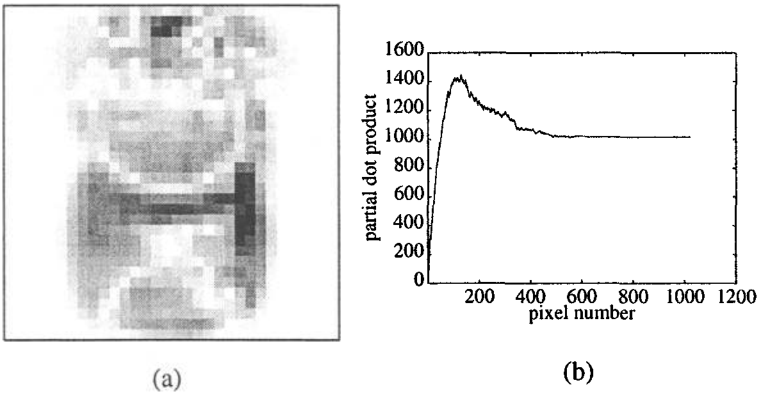

Fig. 6. Relative weights of the components of the normal vector $\mathbf{w}$. See text for details.

In conclusion the proposed method performs recognition with excellent percentages of success even in the presence of very similar objects. It is worthwhile noticing that while the recognition time is practically negligible (requiring the evaluation of 31 dot products), the training stage (in which all the $32 \times 31/2 = 496$ OSHs must be determined) takes about 15 minutes on a SPARC10 workstation.

4.2 Robustness

In order to verify the effectiveness and robustness of the proposed recognition system, we performed experiments under increasingly difficult conditions: pixelwise random noise, bias in the registration, and small amounts of occlusion.

Noise corrupted images We added zero mean random noise to the grey value of each pixel and rescaled the obtained grey levels between 0 and 255. Restricting the analysis to the 32 objects most difficult to recognize, the system performed equally well for maximum noise up to ±100 grey levels and degrades gracefully for higher percentages of noise (see Table 1). Some of the noise corrupted images from which the system was able to identify the viewed object are displayed in Figure 7.

Table 1. Average overall error rates for noise corrupted images. The noise is in grey levels.

Noise	±25	±50	±75	±100	±150	±200	±250
32 Objects	0.3%	0.8%	1.1%	1.6%	2.7%	6.2%	11.0%
30 Objects	0.0%	0.1%	0.2%	0.2%	0.7%	1.8%	5.8%

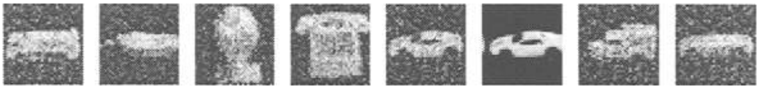

Fig. 7. Eight images synthetically corrupted by white noise, spatially misregistrated and their combination. All these images were correctly classified by the system.

By inspection of the obtained results, we noted that most of the errors were due the three chewing gum packets of Figure 2 which become practically indistinguishable as the noise increases. The same experiments leaving out two of the three packets produced much better performances (see rightmost column of Table 1). It must be said that the very good statistics of Table 1 are partly due to the "filtering effects" of the reduction of the image size from 128 × 128 to 32 × 32 pixels obtained by spatial averaging.

From the obtained experimental results, it can be easily inferred that the method achieves very good recognition rates even in the presence of large amount of noise.

Shifted images We checked the dependence of the system on the precision with which the available images are spatially registered. We thus shifted each image of the test set by n pixels in the horizontal direction and repeated the same recognition experiments of this section on the set of the 32 most difficult objects. As can be appreciated from Table 2, the system performs equally well for small shifts ($n = 3, 5$) and degrades slowly for larger displacements ($n = 7, 10$).

We have obtained very similar results (reported in [9]) when combining noise and shifts. It is concluded that the spatial registration of images is important but

Table 2. Average overall error rates for shifted images (the shifts are in pixel units).

Shift	3	5	7	10
32 Objects	0.6%	2.0%	6.7%	18.6%
30 Objects	0.1%	0.8%	4.8%	12.5%

that spatial redundancy makes it possible to achieve very good performances even in the presence of a combination of additive noise and shift. Here again it must be noted that the quality of the results is partly due to the "filtering effects" of the preprocessing step.

Occlusions In order to verify the robustness of the system against occlusions we performed two more series of experiments. In the first series we randomly selected a subwindow in the rescaled test images (32×32) and assigned a random value between 0 and 255 to the pixels inside the subwindow. The obtained error rates are summarized in Table 3. In the second experiment we randomly selected n columns and m rows in the rescaled images and assigned a random value to the corresponding pixels. The obtained error rates are summarized in Table 4. Some of the images from which the system was able to identify partially occluded objects are displayed in Figure 8. Comparing the results in Tables 3 and 4 it is evident that the occlusion concentrated in a subwindow of the image poses more problems. In both cases, however, we conclude that the system tolerates small amounts of occlusion.

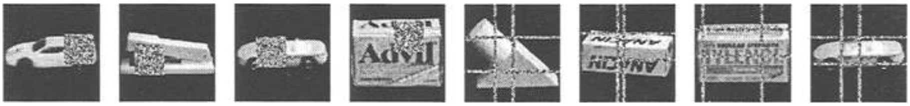

Fig. 8. Eight images with small occlusions correctly classified by the system.

Table 3. Average overall error rates for images occluded by squared window of k pixel per edge.

k	4	6	8	10
32 Objects	0.7%	2.0%	5.7%	12.7%
30 Objects	0.4%	1.2%	4.3%	10.8%

Table 4. Average overall error rates for images occluded by n columns and m rows.

n	m	32 objects	30 objects
1	1	2.1%	1.3%
1	2	3.2%	1.9%
2	1	4.5%	2.8%
2	2	6.1%	3.2%

4.3 Comparison with perceptrons

In order to gain a better understanding of the relevance of the obtained results we run a few experiments using perceptrons instead of SVMs. We considered two objects (the first two toy cars in Figure 2) and run the same experiments described in this section. The results are summarized in table 5. The perceptron column gives the average of the results obtained with ten different perceptrons (corresponding to 10 different random choices of the initial weights). The poor performance of perceptrons can be easily explained in terms of the margin associated with the separating hyperplane of each perceptron as opposed to the SVM margin. In this example, the perceptron margin is between 2 and 10 times smaller than the SVM margin. This means that both SVMs and perceptrons separate exactly the training set, but that the perceptron margin makes it difficult to classify correctly novel images in the presence of noise. Intuitively, this fact can be explained by thinking of noise perturbation as a motion in object space: if the margin is too small, even a slight perturbation can bring a point across the separating hyperplane (see Figure 1).

Table 5. Comparison between SVMs and perceptrons in the presence of noise.

Noise	±50	±100	±150	±200	±250	±300
SVM	0.0%	0.0%	0.0%	0.0%	0.1%	4.1%
Mean Perc.	2.6%	7.1%	15.5%	23.5%	30.2%	34.7%

5 Discussion

In this final section we compare our results with the work of [5] and summarize the obtained results.

The images of the COIL database were originally used by Murase and Nayar as a benchmark for testing their appearance-based recognition system. Our results seem to compare favorably with respect to the results reported in [5] especially in terms of computational cost. This is not surprising because thanks to the design of SVMs, we make use of all the available information with no need of data reduction. Note that SVMs allow for the construction of training sets of much smaller size than the training sets of [5]. Unlike Murase and Nayar's method, however, our method does not identify object's pose.

It would be interesting to compare our method with the classification strategy suggested in [5] on the same data points. After the construction of parametric eigenspaces, Murase and Nayar classify an object by computing the minimum of the distance between the point representative of the object and the manifold of each object in the database. A possibility could be the use of SVMs for this last stage.

In conclusion, in this paper we have assessed the potential of linear SVMs in the problem of recognizing 3–D objects from a single view. As shown by the comparison with other techniques, it appears that SVMs can be effectively *trained* even if the number of examples is much lower than the dimensionality of the object space. This agrees with the theoretical expectation that can be derived by means of VC-dimension considerations [12]. The remarkably good results which we have reported indicate that SVMs are likely to be very useful for direct 3–D object recognition, even in the presence of small amounts of occlusion.

References

1. Bazaraa, M., Shetty, C.M.: Nonlinear programming. (John Wiley, New York, 1979).
2. Brunelli, R., Poggio, T.: Face Recognition: Features versus Templates. IEEE Trans. on PAMI, **15** (1993) 1042–1052
3. Cortes C., Vapnik, V.N.: Support Vector Network. Machine learning **20** (1995) 1–25
4. Edelman, S., Bulthoff, H., Weinshall, D.: Stimulus Familiarity Determines Recognition Strategy for Novel 3–D Objects. AI Memo No. 1138, MIT, Cambridge (1989)
5. Murase, N., Nayar, S.K.: Visual Learning and Recognition of 3–D Object from Appearance. Int. J. Comput. Vision **14** (1995) 5–24
6. Osuna, E., Freund, R., Girosi, F.: Training Support Vector Machines: an Applications to Face Detection. Proc. Int. Conf. Computer Vision and Pattern Recognition, Puerto Rico, (1997)
7. Poggio, T., Edelman, S.: A Network that Learns to Recognize Three-Dimensional Objects. Nature **343** (1990) 263–266
8. Pontil, M., Verri, A.: Properties of Support Vector Machines. Neural Computation **10** (1998) 977–966
9. Pontil, M., Verri, A.: Support Vector Machines for 3-D Objects Recognition. IEEE Trans. on PAMI (to appear)
10. Tarr, M., Pinker, S.: Mental Rotation and Orientation-Dependence in Shape Recognition. Cognitive Psychology **21** (1989) 233–282
11. Vapnik, V.N., Chervonenkis, A.J.: On the uniform convergence of relative frequencies of events to their probabilities. Theory Probab Appl. **16** (1971) 264–280
12. Vapnik, V.N.: The Nature of Statistical Learning Theory. (Springer-Verlag, New York, 1995).