Face Recognition with Multilevel B-Splines and Support Vector Machines

Manuele Bicego
Dipartimento di Informatica
University of Verona
Strada Le Grazie 15
37134 Verona - Italia
bicego@sci.univr.it

Gianluca Iacono
Dipartimento di Informatica
University of Verona
Strada Le Grazie 15
37134 Verona - Italia
iacono@sci.univr.it

Vittorio Murino
Dipartimento di Informatica
University of Verona
Strada Le Grazie 15
37134 Verona - Italia
vittorio.murino@univr.it

ABSTRACT

This paper presents a new face recognition system, based on Multilevel B-splines and Support Vector Machines. The idea is to consider face images as heightfields, in which the height relative to each pixel is given by the corresponding gray level. Such heightfields are approximated using Multilevel B-Splines, and the coefficients of approximation are used as features for the classification process, which is performed using Support Vector Machines. The proposed approach was thoroughly tested, using ORL, Yale, Stirling and Bern face databases. The obtained results are very encouraging, outperforming traditional methods like eigenface, elastic matching or neural-networks based recognition systems.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*; I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Design, Performance

Keywords

Face recognition, Support Vector Machines, Multi Level B-splines

1. INTRODUCTION

Face recognition is undoubtedly an interesting research area, of increasing importance in recent years, due to its applicability as a biometric system in commercial and security applications. These systems could be used to prevent unauthorized access or fraudulent use of ATMs, cellular phones, smart cards, desktop PCs, workstations, and computer networks. The face recognition system has the appealing characteristic of not being an invasive control tool, as compared with fingerprint or iris biometric systems.

A large literature is available on this topic: the first approaches, in the 70's, were based on geometric features [10]. In [3], *features-based matching* and *template matching* methods were compared. One of the best known face recognition method is the so-called *Eigenface* method [26, 28, 20, 2, 30], which uses the *Principal Component Analysis* [8] to project faces into a low-dimensional space, where every face can be expressed as a linear combination of the eigenfaces. This method is not robust against variations of the face orientation and one solution was given by the view-based eigenspace method introduced in [22]. Another important approach is *Elastic Matching* [30, 14, 27, 13], introduced to obtain invariance against expression changes. The idea is to build a lattice on image faces (rigid matching stage), and calculate at each point of the lattice a bank of Gabor filters. In case of variations of expression, this lattice can warp to adapt itself to the face (elastic matching stage). Many other methods have been proposed in the last decade, using different techniques, such as Neural Networks [5, 18, 16], or Hidden Markov Models [12, 24, 21, 6]. Recently, *Independent Component Analysis* was used to project faces into a low-dimensional space, similar to Eigenfaces [29]. Koh *et al.* [11] use a radial grid mapping centered on the nose to extract a feature vector: in correspondence of each point of the grid, the mean value of a circular local patch is calculated and forms an element of the feature vector. Then, the feature vector is classified by a radial basis function neural network. Ayinde *et al.* [1] apply Gabor filters of different sizes and orientations on face images using rank correlation for classification.

In this paper, a different approach is proposed. We consider the face image as a heightfield, in which the height relative to each pixel is given by the corresponding gray level. This surface is approximated using Multilevel B-Splines [17], an interpolation and approximation technique for scattered data. The resulting approximation coefficients were used as features for the classification, carried out by the Support Vector Machines (SVM) [4]. This classifier has already been applied to the face recognition problem [7], in order to classify Principal Components of faces, obtaining very promising results. Moreover, the use of the Support Vector Machines in the context of face authentication has been investigated

WBMA'03, November 8, 2003, Berkeley, California, USA.

in [9].

The reasons underlying the choice of using Multilevel B-Splines and Support Vector Machines are the following: from one hand, Multilevel B-Splines coefficients have been chosen for their approximation capabilities, able to manage slight changes in facial expression. On the other hand, even if a considerable dimensionality reduction is obtained by this technique with respect to considering the whole image, the resulting space is still large. Standard classifiers could be affected by the so called curse of dimensionality problem; SVMs, instead, are well suited to work in very high dimensional spaces (see for example [23]).

The proposed approach was thoroughly tested using most popular databases, such as ORL, Yale, Stirling and Bern[1], and compared with several different approaches. As shown in the experimental section, results obtained are very encouraging, outperforming traditional methods like eigenface, elastic matching and neural-networks based recognition system. Classification accuracies of our approach also outperform those proposed in [7], where SVMs are used with PCA coefficients as features, showing that Multilevel B-splines are very effective and accurate features, able to properly characterize face images.

The rest of the paper is organized as follows. Section 2 contains theoretical background about Multilevel B-Splines and Section 3 is dedicated to *Support Vector Machines* description. In Section 4 the proposed strategy is detailed and experimental results, including a comparative analysis with different methods and several face databases, are reported in Section 5. In Section 6 conclusions are finally drawn.

2. MULTILEVEL B-SPLINES

The *Multilevel B-Splines* [17] represent an approximation and interpolation technique for scattered data. More formally, let $\Omega = \{(x, y)|0 \le x \le m, 0 \le y \le n\}$ be a rectangular non-integer domain in the xy plane. Consider a set of scattered data points $P = \{(x_c, y_c, z_c)\}$ in 3D space, where (x_c, y_c) is a point in Ω. The *approximation function* f is defined as a regular B-Spline function, defined by a control lattice Φ overlaid to Ω, visualized in Fig. 1. Let Φ be a $(m + 3) \times (n + 3)$ lattice that spans the integer grid Ω.

The *approximation B-Spline function* is defined in terms of these control points by:

$$f(x, y) = \sum_{k=0}^{3}\sum_{l=0}^{3} B_k(s)B_l(t)\phi_{(i+k)(j+l)} \quad (1)$$

where $i = \lfloor x\rfloor - 1$, $j = \lfloor y\rfloor - 1$, $s = x - \lfloor x\rfloor$, $t = y - \lfloor y\rfloor$, ϕ_{ij} are control points, obtained as weighted sums with B-Spline coefficients B_k and B_l of 4×4 set of points, called proximity sets, belonging to Ω:

$$\phi_{ij} = \frac{\sum_c w_c^2 \phi_c}{\sum_c \omega_c^2} \quad (2)$$

where $\omega_c = \omega_{kl} = B_k(s)B_l(t)$, $k = (i + 1) - \lfloor x_c\rfloor$, $l = (j+1)-\lfloor y_c\rfloor$, $s = x_c - \lfloor x_c\rfloor$, $t = y_c - \lfloor y_c\rfloor$, (x_c, y_c, z_c) control points and $\phi_c = \frac{w_c z_c}{\sum_{a=0}^{3}\sum_{b=0}^{3} w_{ab}^2}$. By properly choosing the resolution of the control lattice Φ, it is possible to obtain a compromise between the precision and smoothness of the function; a good smoothness entails a cost in terms of low accuracy, and vice-versa.

[1]Downloadable respectively from:
http://www.uk.research.att.com/facedatabase.html
http://cvc.yale.edu/projects/yalefaces/yalefaces.html
http://pics.psych.stir.ac.uk
ftp://iamftp.unibe.ch/pub/Images/FaceImages.

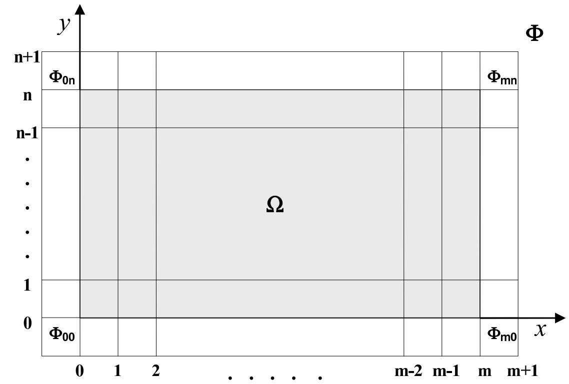

Figure 1: Configuration of control lattice Φ in relation to domain Ω.

Multilevel B-Splines approximation can overcome this problem. Consider a hierarchy of control lattices $\Phi_0, \Phi_1, \ldots, \Phi_h$, that spans the domain Ω. Assume that, having fixed the resolution of Φ_0, the spacing between control points in Φ_i is halved from one lattice to the next.

The process of approximation starts by applying the basic B-Spline approximation to P with the coarsest control lattice Φ_0, obtaining a smooth initial approximation f_0. f_0 leaves a deviation $\Delta^1 z_c = z_c - f_0(x_c, y_c)$ for each point (x_c, y_c, z_c) in P. Then, f_1 is calculated by the control lattice Φ_1, approximating the difference $P_1 = \{(x_c, y_c, \Delta_c^1)\}$. The sum $f_1 + f_2$ yields a smaller deviation $\Delta^2 z_c = z_c - f_0(x_c, y_c) - f_1(x_c, y_c)$ for each point (x_c, y_c, z_c) in P.

In general, for every level k in the hierarchy, using the control lattice Φ_k, a function f_k is derived to approximate data points $P_k = \{(x_c, y_c, \Delta^k z_c)\}$, where $\Delta^k z_c = z_c - \sum_{i=0}^{k-1} f_i(x_c, y_c)$, and $\Delta^0 z_c = z_c$. This process starts with the coarsest control lattice Φ_0 up to the highest lattice Φ_h. The final function f is calculated by the sum of functions f_k, $f = \sum_{k=0}^{h} f_k$.

In general, the higher the resolution of the coarsest control lattice Φ_0, the lower the smoothness of the final function. This behavior is exemplified in Fig. 2, where different approximating functions, built with different starting coarser lattices, are shown. Given a set of points in a domain $width \times height$, m and n indicate that the lattice Φ, on which the approximating function has been built, has dimension $\left(\lfloor \frac{width}{m}\rfloor + 3\right) \times \left(\lfloor \frac{height}{n}\rfloor + 3\right)$. It follows that high values of m and n indicate low dimensions of Φ.

In the basic Multilevel B-Splines algorithm, the evaluation of f involves the computation of the function f_k for each level k, summing them over domain Ω (Fig. 3(a)). This introduces a significant overhead in computational time, if f has to be evaluated at a large number of points in Ω. To address this problem, Multilevel B-Splines refinement has been proposed in [17]. This technique allows to represent f by one B-Spline function rather than by the sum of several B-Spline functions.

Let $F(\Phi)$ be the B-spline function generated by control lattice Φ and let $|\Phi|$ denote the size of Φ. With B-spline refinement, we can derive the control lattice Φ'_0 from the coarsest lattice Φ_0 such that $F(\Phi'_0) = f_0$ and $|\Phi'_0| = |\Phi_1|$.

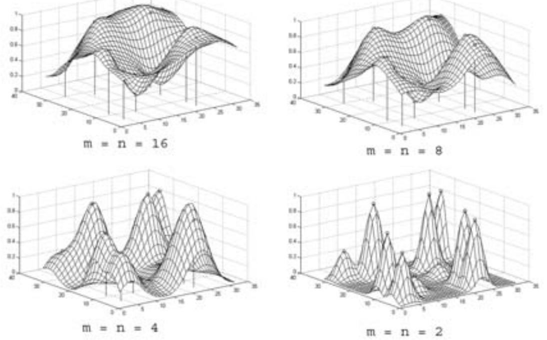

Figure 2: Examples of Multilevel B-Splines approximation, using different resolutions (m and n) of the first control lattice Φ_0.

Then, the sum of functions f_0 and f_1 can be represented by control lattice Ψ_1 which results from the addition of each corresponding pair of control points in Φ_0' and Φ_1. That is, $F(\Psi_1) = g_1 = f_0 + f_1$, where $\Psi_1 = \Phi_0' + \Phi_1$.

In general, let $g_k = \sum_{i=0}^{k} f_i$ be the partial sum of functions f_i up to level k in the hierarchy. Suppose that function g_{k-1} is represented by a control lattice Ψ_{k-1} such that $|\Psi_{k-1}| = |\Phi_{k-1}|$. In the same manner as we computed Ψ_1 above, we can refine Ψ_{k-1} to obtain Ψ_{k-1}' , and add Ψ_{k-1}' to Φ_k to derive Ψ_k such that $F(\Psi_k) = g_k$ and $|\Psi_k| = |\Phi_k|$. That is, $\Psi_k = \Psi_{k-1}' + \Phi_k$. Therefore, from $g_0 = f_0$ and $\Psi_0 = \Phi_0$, we can compute a sequence of control lattices Ψ_k to progressively derive control lattice Ψ_h for the final approximation function $f = g_h$. A scheme of this procedure is shown in Fig. 3(b).

3. SUPPORT VECTOR MACHINES

Support Vector Machines [4] are binary classifiers, able to separate two classes through an optimal hyperplane. The optimal hyperplane is the one maximizing the "margin", defined as the distance between the closest examples of different classes. To obtain a non-linear decision surface, it is possible to use *kernel functions*, in order to project data in a high dimensional space, where a hyperplane can more easily separate them. The corresponding decision surface in the original space is not linear.

The rest of this section details the theoretical and practical aspects of Support Vector Machines: firstly, linear SVMs are introduced, for both linearly and not linearly separable data. Subsequently, we introduce non linear SVMs, able to produce non linear separation surfaces. A very useful and introductory tutorial on Support Vector Machines for Pattern Recognition can be found in [4].

In the case of linearly separable data, let $D = \{(\mathbf{x_i}, y_i)\}, i = 1 \dots \ell, y_i \in \{-1, +1\}, \mathbf{x_i} \in \Re^d$ be the *training set* of the SVMs. D is linearly separable if exists $\mathbf{w} \in \Re^d$ and $b \in \Re$, such that:

$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) \geq 1 \text{ for } i = 1, \dots, \ell \quad (3)$$

$H : \mathbf{w} \cdot \mathbf{x} + b = 0$ is called the "separating hyperplane". Let $d_+(d_-)$ be the minimum distance of the separating hyperplane from the closest positive (negative) point. Let us define the "margin" of the hyperplane as $d_+ + d_-$. Different

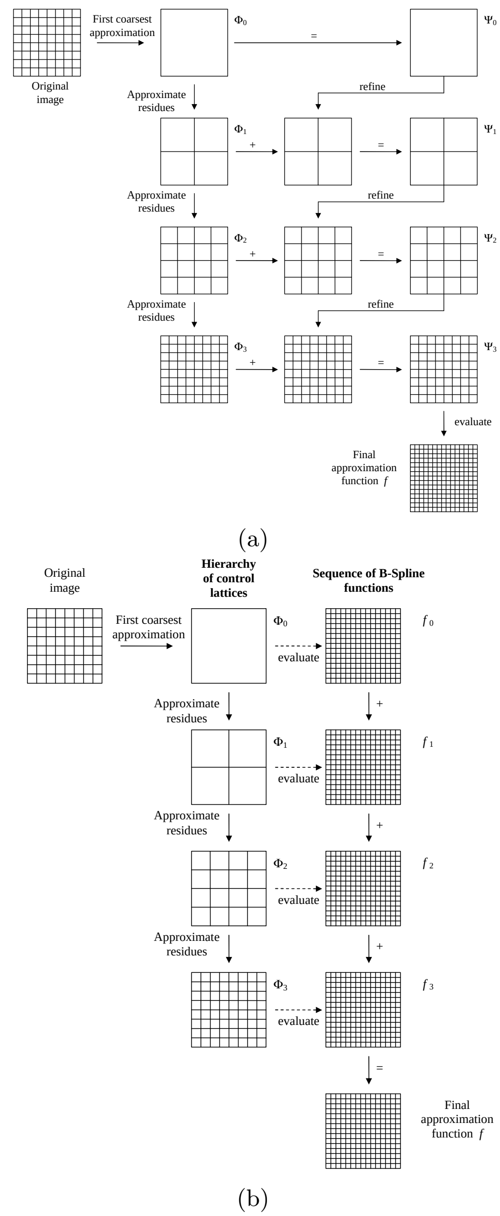

Figure 3: Description of MBA algorithm: (a) basic version; (b) MBA with refinement.

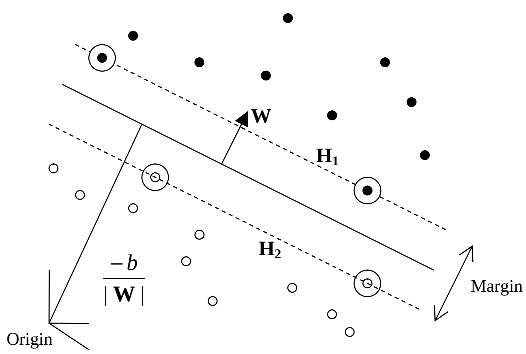

Figure 4: Geometric interpretation of SVMs. A hyperplane separates black points from white points. The hyperplane is obtained as a linear combination of the circled points, called *support vectors*, and is defined by a direction vector W and a distance-from-origin scalar b.

separating hyperplanes exist. SVMs find the one that maximizes the margin. Let us define $H_1 : \mathbf{w} \cdot \mathbf{x} + b = +1$ and $H_2 : \mathbf{w} \cdot \mathbf{x} + b = -1$. The distance of a point of H_1 from $H : \mathbf{w} \cdot \mathbf{x} + b = 0$ is $\frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$, and the distance between H_1 and H_2 is $\frac{2}{\|\mathbf{w}\|}$. So, to maximize the margin, we must minimize $\|\mathbf{w}\| = \mathbf{w}^T \mathbf{w}$, with the constraints that no points lie between H_1 and H_2.

It can be proven [4] that the problem of training a SVM is reduced to the solution of the following Quadratic Programming (QP) problem:

$$\max\{-\frac{1}{2}\alpha^T B \alpha + \sum_{i=1}^{\ell} \alpha_i\} \tag{4}$$

$$\sum_{i=1}^{\ell} y_i \alpha_i = 0 \text{ and } \alpha_i \geq 0 \tag{5}$$

where α_i are Lagrange coefficients and B is a $\ell \times \ell$ matrix defined as:

$$B_{ij} = y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} \tag{6}$$

The optimal hyperplane is determined with $\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x_i}$, and the classification of a new point $\mathbf{x}$ is obtained by calculating $sgn(\mathbf{w} \cdot \mathbf{x} + b)$. It is important to observe that only those $\mathbf{x_i}$ whose corresponding Lagrange coefficients α_i are not null contribute to the sum that defines the separating hyperplane. For this reason, these points are called *support vectors* and, geometrically, lie along the two hyperplanes H_1 and H_2 (see the Fig. 4). When data points are not linearly separable, slack variables are introduced, in order to allow points to exceed margin borders:

$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) \geq 1 - \xi_i \tag{7}$$

The idea is to permit such situations, by controlling them by the introduction of a cost parameter C. This parameter determines the sensibility of the SVM to classification errors: a high value of C strongly penalizes errors, also at the cost of a narrow margin, while a low value of C permits some classification errors. Intermediate values of C result in a compromise between the minimization of the number of errors and maximization of the margin. Finally, the training process results in the solution of the following QP problem:

$$\max \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} \tag{8}$$

$$\sum_{i=1}^{\ell} y_i \alpha_i = 0 \text{ and } 0 \leq \alpha_i \leq C \tag{9}$$

The SVM approach could also be generalized to the case where the decision function is not a linear function of the data: in this case we have the so-called non-linear SVM. The idea under nonlinear SVMs is to project data points into a high, even huge, dimensional Hilbert space H, by using a function Ξ such that:

$$\begin{aligned} \Xi : \Re^d &\rightarrow H \\ \mathbf{x} &\rightarrow z(\mathbf{x}) = \mathbf{z}(\xi_1(\mathbf{x}), \xi_2(\mathbf{x}), \ldots, \xi_n(\mathbf{x})) \end{aligned}$$

and then separate projected data points through a hyperplane.

First of all, notice that the only way in which the data appear in the training problem is in the form of inner products $\mathbf{x_i} \cdot \mathbf{x_j}$. When projecting points $\mathbf{x}$ in $\Xi(\mathbf{x})$, the training process will still depend on the inner product of projected points $\Xi(\mathbf{x_i}) \cdot \Xi(\mathbf{x_j})$. Then, to solve the problem of nonlinear decision surfaces, it is sufficient to modify the training and classification algorithms, substituting the inner product between data points of the training set with a *kernel* function K, such that:

$$K(\mathbf{x_i}, \mathbf{x_j}) = \Xi(\mathbf{x_i}) \cdot \Xi(\mathbf{x_j}) \tag{10}$$

To be a *kernel*, a function must verify *Mercer conditions* [4]. Some examples of *kernel* are *polynomial functions* like $K(x, y) = ((x \cdot y) + 1)^d$, *exponential radial basis function* and *multi-layer perceptron*. In this way, data points are projected in a higher dimensional space, where a hyperplane could be sufficient to separate the problem properly. It is important to notice that, by the use of this "kernel trick", the non linear decision surface is obtained in roughly the same amount of time needed to build a linear SVM.

4. THE STRATEGY

In this section, the proposed strategy is detailed: features are extracted using Multilevel B-Splines, and successively classified using Support Vector Machines.

First, the face image should be sampled, in order to obtain a set of points to approximate. This set is obtained by the fusion of two subsets: firstly, a Canny filter is applied to extract edges from the image faces. Therefore, the first subset of control points to approximate is $P_1 = (x_{c_1}, y_{c_1}, z_{c_1})$, with x_{c_1}, y_{c_1} coordinates of the edges and z_{c_1} the corresponding gray levels. The second subset of control points is given by $P_2 = (x_{c_2}, y_{c_2}, z_{c_2})$, with x_{c_2}, y_{c_2} coordinates corresponding to a sub-sampling carried out on the image. Finally, the set of control points to approximate is given by $P = P_1 \cup P_2$, as shown in Fig. 5(b) as an example. Subsequently, the approximation algorithm is applied to this set of points, considering the control lattice coefficients as features. Once extracted, the control lattice is linearized into a feature vector, using the standard raster scan.

Face recognition is a multi-class classification problem, but Support Vector Machines are binary classifiers. To ex-

Figure 5: Feature extraction stage. Original face image and used control points.

tend SVMs to the multi-class case, we adopted the strategy of binary decision trees proposed by Verri *et al.* [23], also called strategy of the tennis tournament, also adopted by Guo *et al.* in their paper [7].

Let us assume to have c classes. The training stage consists in building up all possible SVMs 1-vs-1[2], combining all the available classes. The number of possible (not ordered) pairs of classes is $\frac{c(c-1)}{2}$. In this way, $\frac{c(c-1)}{2}$ SVMs are trained. In the classification stage, a binary decision tree is built, starting from the leaves, in which each pair of brother nodes represent a SVM. Given a test image, recognition was performed following the rules of a tennis tournament. Each class is regarded as a player, and in each match the system classifies the test images according to the decision of the SVM of the pair of players involved in the match. The winner identities, proposed by each SVM, will be propagated to the upper level of the tree, playing again. The process continues until the root is reached. Finally, the root will be labelled with the identity of the classified subject. Because it is *a priori* impossible to know which SVM will define the various levels of the tree, the necessity of training all possible SVMs 1-vs-1 is now clear.

In Fig. 6, an example of this classification rule is proposed. In principle, different choices of the starting configuration, regarding SVMs inserted as leaves, could lead to different results. Nevertheless, in practice, preliminary experiments showed that averaged accuracies do not depend from the starting configuration.

If c does not equal to the power of 2, we can decompose c as: $c = 2^{n_1} + 2^{n_2} + \ldots + 2^{n_I}$, where $n_1 \geq n_2 \geq \ldots \geq n_I$. If c is an odd number, $n_I = 0$; otherwise, $n_I > 0$. Then, we can build I trees, the first with n_1 leaves, the second with n_2 and so on. Finally, starting from the I roots, we can build the final tree (or, if necessary, recursively decompose I again in powers of 2). Even if this decomposition is not unique, the number of comparisons in the classification stage is always $c - 1$.

5. EXPERIMENTAL RESULTS

In this section, experimental results are proposed. We preliminary studied three different types of approximation coefficients as features, regarding three different methods

[2]We call this kind of SVMs 1-vs-1, in order to distinguish them from SVMs 1-vs-all, that were trained to classify between faces of one class and faces of all other classes.

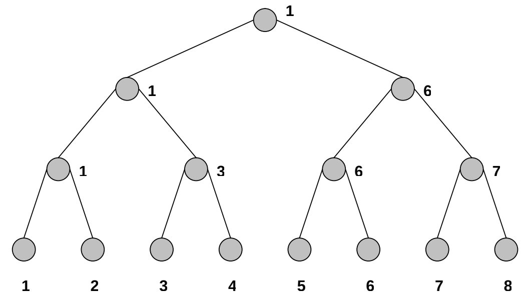

Figure 6: An example of multi-class classification.The subject to be recognized belongs to class number 1. First, it is classified by the SVM relative to classes 1-2, 3-4, 5-6, 7-8. The winners of this first set of classifications will define the upper level of the tree, constituted by SVMs relative to pairs 1-3 and 6-7. Finally, the final SVM relative to classes 1 and 6 establishes the winner.

Database	N.subj	N.*training*	N.*testing*
ORL	40	5	5
Yale	15	5	6
Stirling	32	3	3
Bern	30	5	5
Combined	117	Variable	Variable

Table 1: Description of the databases used for experiments: number of subjects, photos per subject used for training and for testing.

to approximate points: B-splines, Multilevel B-splines, and Multilevel B-splines with refinement. The features we investigated were therefore: the ϕ_{ij}'s calculated with the B-Spline approximation method under different resolutions, the ϕ_{ij}'s of the control lattice hierarchy calculated with the Multilevel B-Splines algorithm, and the ψ_{ij}'s calculated with the Multilevel B-Splines algorithm with refinement. This last algorithm obtained the best results, in terms of recognition rate: therefore, in the following, only results regarding this algorithm are proposed.

About the kernel used in Support Vector Machine, after several experimental tests, the best performance was reached by the *Exponential Radial Basis Function*, which provided the most stable results. In our experiments the parameters of the SVMs were set to $C = 100$ and $\sigma = 20$, with σ the kernel variance, being the results stable for $C \geq 100$ and $\sigma \geq 20$.

We used different face databases to test the proposed system. Databases, number of subjects and photos per subjects used in the training and testing set, are shown in Table 1. Moreover, in this table, "combined" means a database obtained by the fusion of all other databases. We tested different random combinations of training and testing sets and results were averaged. All face images were resized to the dimensions of ORL databases photos, i.e. 92×112 pixel.

Some comments about the databases used: ORL database contains a high within-class variance, like illumination changes, facial expressions, glasses/no glasses, and scale. Subjects in the Yale database are characterized by high illumination

Level	MBA-REF	BA
32	4.25%	4.25%
16	2.75%	3.13%
8	3.75%	4%
4	3.88%	4.25%
2	4.13%	4.13%

(a)

Database	Errors
Yale	1.11%
Stirling	1.04%
Bern	4.67%
Combined	3.3%

(b)

Table 2: Recognition error rates: (a) on ORL database, with different resolutions of the control lattice used as features vector. MBA-REF stands for Multilevel B-Splines approximation with refinement and BA stands for basic B-Splines approximation; (b) on other databases, with Multilevel B-Splines with refinement and level 16.

changes, presence and absence of glasses, and variations in facial expressions. Stirling database contains subjects in various poses and expressions (smiling and speaking). Finally, Bern database presents photos under partially controlled illumination conditions, but with different poses.

In Table 2(a), classification error rates calculated on ORL face database are shown. With the aim of understanding how to exploit the multi-level nature of the described algorithms, the system performances on this database are proposed for different levels of resolution of the control lattice. We recall that "level n" means dimensionality of the control lattice Φ equal to $\left(\lfloor\frac{height}{n}\rfloor+3\right)\times\left(\lfloor\frac{width}{n}\rfloor+3\right)$, where *height* and *width* are relative to image dimensions. Moreover, in Table 2(a) a comparison between coefficients of Multilevel B-Splines with refinement and basic B-Spline approximation as features is shown. It can be noticed how the formers obtain a better performance.

One can note that results are very satisfactory, confirming that our approach is accurate and effective. It can also be noticed how, after level 16, performances get worse, probably due to a problem of *over-fitting*, and the same behavior has been noted also on other databases, even if results are not reported here.

With 92×112 pixel photos, the dimensionality of the control lattice, corresponding to the level 16, equals to $\left(\lfloor\frac{92}{16}\rfloor+3\right)\times\left(\lfloor\frac{112}{16}\rfloor+3\right)=80$.Considering that images contain $92\times112 = 10304$ pixels, level 16 permits a really noticeable dimensionality reduction, equal to about two orders of magnitude, precisely 99,22%.

The recognition error rates computed on Yale, Stirling, Bern and combined databases are shown in Table 2(b), only for resolution level equal to 16 (best results). Also in this case, errors rates are very low, nearly to a perfect classification for the Yale and Stirling databases.

Some comparative results on the ORL database are reported in Table 3, where our method is named MBA+SVM. We can note that our approach is highly competitive: only two results are substantially better than ours, obtained from the two approaches using HMM and DCT coefficients [12, 6]. Our method outperforms standard approaches like eigenfaces, neural network and elastic matching. Also the method that uses SVM with PCA features [7] is outperformed, showing that Multilevel B-splines coefficients are effective and accurate features, able to properly model faces. Moreover Multilevel B-Splines coefficients show a better discrimination accuracy when coupled with SVMs that, we recall, in high-discriminating feature spaces suffer the problem of over-training.

Method	Error	Ref.	Year
Top-down HMM + gray tone features	13%	[25]	1994
Eigenface	9.5%	[28]	1994
Pseudo 2D HMM + gray tone features	5.5%	[24]	1994
Elastic matching	20.0%	[30]	1997
PDNN	4.0%	[18]	1997
Continuous n-tuple classifier	2.7%	[19]	1997
Top-down HMM + DCT coef.	16%	[21]	1998
Point-matching and correlation	16%	[15]	1998
Ergodic HMM + DCT coef.	0.5%	[12]	1998
Pseudo 2D HMM + DCT coef.	0%	[6]	1999
SVM + PCA coef.	3%	[7]	2001
Indipendent Component Analysis	15%	[29]	2002
Gabor filters + rank correlation	8.5%	[1]	2002
SVM + MBA coef.	2.75%		2003

Table 3: Comparative results on ORL database. SVM + MBA stands for our method.

Method	Error
Eigenface	13%
Elastic Matching	7%
Back Propagation Neural Networks	57%
MBA+SVM	4.67%

Table 4: Comparative results on Bern databases.

Other comparative results are obtained on Bern database from [30], regarding eigenfaces, elastic matching and neural networks. The comparison is shown in Table 4: also in this case our method reached better results.

In Fig. 7, some photos of typical misclassified subjects are shown. It is interesting to note that the subjects in the Fig. show a certain degree of similarity, for the presence, for example, of the beard or glasses.

It has been observed that, when a misclassification occurs, the correct identity tends to go up in the decision tree, up to levels close to the root. Quantitatively, in 82% of the erroneous situations on the ORL database experiment, the correct identity was found in the second level of the decision tree, *i.e.* as children of the root. The query of the database aimed to obtain the k identities closest to the root could further increase the probability of obtaining the right identity

Figure 7: Examples of misclassified subjects. For every pair, the face to be recognized is shown on the left, an example image belonging to the erroneous class stated by the system is shown on the right.

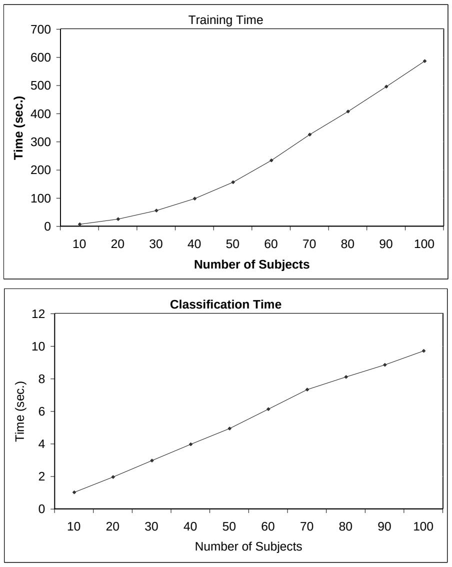

Figure 8: Training and classification times.

in output.

The computational cost of the training and classification procedures requirements are shown in Fig. 8, in function of the number of classes. These rates have been calculated on a Intel Celeron 850 Mhz with 256 Mb RAM, using MATLAB 5.2 routines. We think that better results should be obtained with an optimized implementation in C language. As can be noticed, in a 100 classes problem the time required for training is about 10 minutes; for classification it is about 10 seconds. The time complexity of the training task is $O(n^2)$ and of the classification task is $O(n)$, where n is the number of subjects.

6. CONCLUSIONS

In this paper, we proposed a new approach to the face recognition problem, based on Multilevel B-splines and Support Vector Machines. Considering the face image as an heightfield, we propose to use as features the control lattices of the Multilevel B-Splines approximation of the face surface. We showed that such features are really discriminating, and operate a remarkable reduction of data dimensionality. The performances we reached, compared to many others recognition systems in the literature, proved the superiority of our approach with respect to well-known standard methods, like eigenfaces, elastic matching and neural networks.

7. REFERENCES

[1] O. Ayinde and Y. Yang. Face recognition approach based on rank correlation of gabor-filtered images. *Pattern Recognition*, 35(6):1275–1289, 2002.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.

[3] R. Brunelli and T. Poggio. Face recognition: Features versus template. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(10), October 1993.

[4] C. Burges. A tutorial on support vector machine for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[5] G. W. Cottrell and M. Fleming. Face recognition using unsupervised feature extraction. In *Proc. Int. Neural Network Conf.*, volume 1, pages 322–325, Paris,France, July 1990.

[6] S. Eickeler, S. Mller, and G. Rigoll. Recognition of jpeg compressed face images based on statistical methods. *Image and Vision Computing*, 18:279–287, March 2000.

[7] G. Guo, S. Z. Li, , and C. Kapluk. Face recognition by support vector machines. *Image and Vision Computing*, 19(9-10):631–638, 2001.

[8] I. Jollife. *Principal component analysis*. Springer Verlag, New York, 1986.

[9] K. Jonsson, J. Kittler, Y. P. Li, and J. Matas. Support vector machines for face authentication. In *Proc. of Brit. Machine Vision Conf., Nottingham, UK*, pages 543–553, 1999.

[10] T. Kanade. Picture processing system by computer complex and recognition of human faces, November 1973. doctoral dissertation, Kyoto University.

[11] L. Koh, S. Ranganath, and Y. Venkatesh. An integrated automatic face detction and recognition system. *Pattern Recognition*, 35(6):1259–1273, 2002.

[12] V. V. Kohir and U. B. Desai. Face recognition using DCT-HMM approach. In *Workshop on Advances in Facial Image Analysis and Recognition Technology (AFIART)*, Freiburg, Germany, June 1998.

[13] C. Koutropoulos, A. Tefas, and I. Pitas. Morphological elastic graph matching applied to frontal face authentication under well-controlled and real conditions. In *International Conference on Multimedia Computing and Systems*, volume 2, pages 934–938, 1999.

[14] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, and R. Wurtz. Distortion invariant object recognition in the dinamic link architecture. *IEEE Trans. on Computers*, 42(3):300–311, 1993.

[15] K. M. Lam and H. Yan. An analytic-to-holistic approach for face recognition on a single frontal view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(7):673–686, 1998.

[16] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: a convolutional neural network approach. *IEEE Trans. on Neural Networks*, 8(1):98–113, November 1997.

[17] S. Lee, G. Wolberg, and S. Y. Shin. Scattered data interpolation with multilevel b-splines. *IEEE Trans. on Visualization and Computer Graphics*, 3(3):228–244, July-September 1997.

[18] S. Lin, S. Kung, and L. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Trans. on Neural Networks*, 8(1):114–131, January 1997.

[19] S. M. Lucas. Face recognition with the continuous n-tuple classifier. In *Proceedings of British Machine Vision Conference*, September 1997.

[20] B. Moghaddam, W. Wahid, and A. Pentland. Beyond

eigenfaces: Probabilistic matching for face recognition. In *The 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, April 1998.

[21] A. V. Nefian and M. H. Hayes. Hidden Markov models for face recognition. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2721–2724, Seattle, May 1998.

[22] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. *Computer Vision and Pattern Recognition*, pages 84–91, 1994.

[23] M. Pontil and A. Verri. Support vector machines for 3-d object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:637–646, 1998.

[24] F. Samaria. *Face recognition using Hidden Markov Models*. PhD thesis, Engineering Department, Cambridge University, October 1994.

[25] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, Sarasota, Florida, December 1994.

[26] L. Sirovich and M. Kirby. Low dimentional procedure for the caracterization of human face. *J. Opt. Soc. Am. A*, 4(3):519–524, March 1987.

[27] A. Tefas, C. Kotropoulos, and I. Pitas. Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(7), July 2001.

[28] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[29] P. C. Yuen and J. H. Lai. Face representation using indipendent component analysis. *Pattern Recognition*, 35(6):1247–1257, 2002.

[30] J. Zhang, Y. Yan, and M. Lades. Face recognition: eigenface, elastic matching and neural nets. *Proceedings of the IEEE*, 85(9), September 1997.