3D Face Recognition Using Stereoscopic Vision

U. Castellani, M. Bicego, G. Iacono, and V. Murino

Dipartimento di Informatica, Università di Verona
Ca' Vignal 2, Strada Le Grazie 15
37134 Verona, Italia
{castellani,bicego,murino}@sci.univr.it

Abstract. In this paper a new complete system for 3D face recognition is presented. 3D face recognition presents several advantages against 2D face recognition, as, for example, invariance to illumination conditions. The proposed system makes use of a stereo methodology, that does not require any expensive range sensors. The 3D image of the face is modelled using Multilevel B-Splines coefficients, that are classified using Support Vector Machines. Preliminary experimental evaluation has produced encouraging results, making the proposed system a promising low cost 3D face recognition system.

1 Introduction

Face recognition is undoubtedly an interesting research area, growing in importance in recent years, due to its applicability as a biometric system in commercial and security applications. The face recognition system has the appealing characteristic of not being an invasive control tool, as compared with fingerprint or iris biometric systems.

The most typical approach to face recognition is to analyze 2D face images, and a large literature is available on this topic (for a review see [1], and [2]). The analysis of 2D face has some inherent drawbacks: for example it is not able to distinguish a real face from a picture of a face, since it does not consider depth information. This could represent an awkward problem, especially in the authentication context. Moreover, most part of techniques proposed in the literature suffers from illumination changes problems.

The analysis of 3D images of a face represents a possible solution for both these problems. Although 3D facial analysis has been already applied in some research areas, as compression and synthesis for videoconferencing [3], recognition of faces basing on range images is still weakly addressed in the literature [4,5,6,7,8,9]. More in detail, the first system that analyzes 3D faces was presented in [4]: the method identified facial features points, based on local curvature computed from range images. The face was segmented in convex and concave regions, and features were determined from these regions. No recognition was performed in this system. Gordon [5,6] was the first that realized a recognition system based on range data. He computed geometric features of the sensed surface, integrating some a priori knowledge. Recognition was performed using a template matching

M. Tistarelli, J. Bigun, and E. Grosso (Eds.): Biometrics School 2003, LNCS 3161, pp. 126–137, 2005.

approach and a classification system in the feature space. Another approach was proposed in [7,8], where the 3D information was determined using a coded light approach with two separate sensors. The classification step was performed using an eigenface approach and HMM-based technique. More recently, the same authors present another system [9], that classifies range images, acquired using a multi sensor system. The canonical position is determined from the range images face, and a 3D Haussdorff distance is used for the classification step.

In this paper a new complete system for 3D face recognition is proposed, based on stereoscopic images analysis. The process of stereo reconstruction aims at recovering the 3D structure from a pair of images by searching for *conjugate points*, i.e., points in the left and right images that are projections of the same scene point. The difference between the positions of conjugate points is called *disparity*. Stereo is a well known issue in Computer Vision, to which many articles have been devoted (see [10] for a survey). The system proposed in this paper has a clear advantage with respect to the previously introduced: the acquisition process is fast and entirely low cost. In fact, the 3D information are acquired using two cameras by applying the stereoscopic principles, without any need of particularly expensive range sensors. Furthermore, the stereo setup calibration is very easy and fast and there are different standard implemented methods freely available on the web. This aspect is really important, especially in the view of enlarging the applicability of the biometric technologies to real problems.

The range image obtained by the stereoscopic analysis is approximated using Multilevel B-Splines [11], 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) [12]. 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 [13], 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 [13]). This classification system has been already employed by the authors in the context of 2D face recognition [14]. In this version we explore the possibility to estimate the face surfaces directly from the 3D data obtained by the acquisition system.

The proposed system has been used to collect a set of 90 faces, with 9 subjects (each with 10 face, varying expressions). Classification accuracies on this data set are very encouraging, and make the proposed approach a promising really employable system for face recognition and authentication.

The rest of the paper is organized as follows: in the Sect. 2 the acquisition system is detailed, while in the recognition system is described in Sect. 3. The experimental evaluation is proposed in 4; finally, Sect. 5 contains conclusions and future perspectives.

2 The Acquisition System

Three-dimensional data are obtained from an active stereo system developed at the VIPS (Vision, Image Processing, and Sound) laboratory[1] of the Department of Computer Science (University of Verona). The system is composed by two optical cameras and a overhead projector which illuminates the scene with a salt-and-pepper random texture (Figure 1). Thus, all the surfaces are textured,

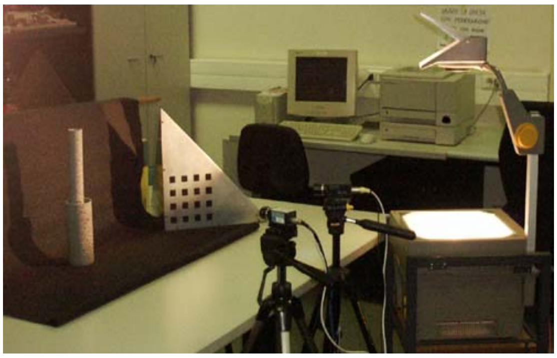


Fig. 1. Active Stereo system of acquisition

and every small surface patch is characterized by a very distinctive pattern (Figure 2(a) and (b)). This trick facilitate area-based stereo matching, which would otherwise produces no meaningful results for uniformly colored areas. In summary, the acquisition pipeline is composed of the following stages:

Calibration. The position and orientation of both cameras, as well as intrinsic parameters are computed with the calibration algorithm described in [15] and implemented in a Matlab toolbox[2].

Rectification. Instead of relying on accurate mechanical alignment, a parallel-camera acquisition geometry is "simulated" by transforming the images captured by the two cameras as if they were taken by two virtual parallel cameras. This process is called *epipolar rectification*, and is described in [16].

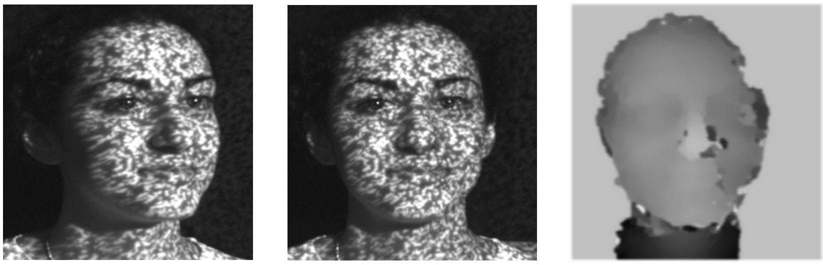
Stereo Matching. Corresponding points on the left and right images are recovered using the R-SMW area-based stereo matching algorithm [17]. However, given that we project an artificial texture onto the scene, the choice of the matching algorithm is less critical than in passive stereo.

The output of the system is a *disparity map* [10] related to the acquired subject (Figure 2(c)). It is worth noting that the disparity map is very similar to a range

[1] See `http://vips.sci.univr.it`.

[2] The Camera Calibration Toolbox for Matlab is downloadable from `http://newbologna.vision.caltech.edu/bouguetj`

map [18] and it covers the 3D information we are using for recognizing the faces. In particular light disparity pixels correspond to surface points that are closer to the sensor and vice versa.



(a) (b) (c)

Fig. 2. Stereo images, left(a) and right (b), acquired while the overhead projector projects a random texture to the subject, and disparity map (c)

3 The Classification System

The classification system is based on two stages: firstly, range images are modelled using Multilevel B-Splines [11] and coefficients of approximation are extracted. Then, these coefficients are used for classification with Support Vector Machines [12].

3.1 Multilevel B-Splines

The *Multilevel B-Splines* [11] represent an approximation and interpolation technique for scattered data. More formally, let $\Omega = \{(x, y)|0 \leq x \leq m, 0 \leq y \leq n\}$ 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. 3. 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:

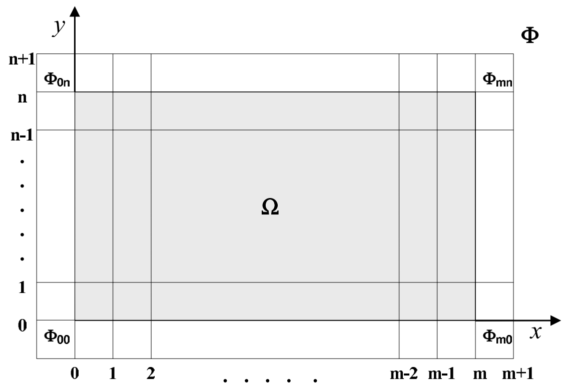


Fig. 3. Configuration of control lattice Φ in relation to domain Ω.

$$\phi_{ij} = \frac{\sum_c w_c^2 \phi_c}{\sum_c \omega_c^2} \tag{2}$$

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.

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. 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. 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 [11]. 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 $|\Phi'_0| = |\Phi_1|$. 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 $f = g_h$.

3.2 Support Vector Machines

Support Vector Machines [12] 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 [12].

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

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

$H : \mathbf{w} \cdot \mathbf{x} + b = 0$ 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 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 [12] 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:

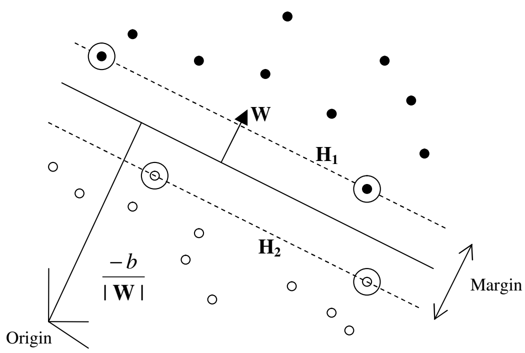


Fig. 4. Geometric interpretation of SVMs. A hyperplane separates black points from white points. The hyperplane is obtained as a linear combination of the circled points, called *support vectors*, and is defined by a direction vector **W** and a distance-from-origin scalar b.

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

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:

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

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* [12]. 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.

3.3 The Classification Strategy

For recognizing 3D faces we have employed the following strategy: firstly, the face surface is sampled, in order to obtain a set of points to approximate. Subsequently, the Multilevel B-Spline Algorithm with refinement (that is a variation to the basic algorithm described in [11]) is applied to this set of points, considering the control lattice coefficients of a certain level 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 extend SVMs to the multi-class case, we adopted the strategy of binary decision trees proposed by Verri *et al.* [13], called strategy of the tennis tournament, also adopted by Guo *et al.* in their paper [19].

Let us assume to have c classes. The training stage consists in building up all possible SVMs 1-vs-1[3], 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. 5, an example of this classification rule is proposed. In principle, different choices of the starting configuration, regarding SVMs inserted as leaves,

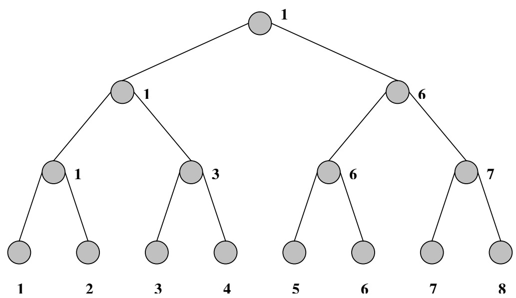


Fig. 5. An example of multi-class classification.The subject to be recognized belongs to class number 1. First, it is classified by the SVM relative to classes 1-2, 3-4, 5-6, 7-8. The winners of this first set of classifications will define the upper level of the tree, constituted by SVMs relative to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

could lead to different results. Nevertheless, in practice, preliminary experiments showed that averaged accuracies do not depend from the starting configuration.

[3] 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.

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 $c - 1$.

4 Experimental Results

The system has been preliminary tested on a set of 9 subjects, each with 10 images, varying expression. Five images were used for the training, while the remaining were used for the testing. The parameters of the approach has been chosen based of a previous analysis on a 2D face recognition problem [14]: the coefficients level of the Multilevel B-splines approximation was set to 16. The SVM was used with the exponential Radial Basis Function kernel, using $\sigma = 10$. The C parameter, which drives the regularization [20], was set to 5. With 150×150 pixel images, the dimensionality of the control lattice, corresponding to the level 16, equals to $\left(\lfloor\frac{150}{16}\rfloor + 3\right) \times \left(\lfloor\frac{150}{16}\rfloor + 3\right) = 144$. Considering that images contain $150 \times 150 = 22.500$ pixels, level 16 permits a really noticeable dimensionality reduction, equal to about two orders of magnitude, precisely 99,36%.

Results are presented in Table 1, for different combination of the training and the testing set. We can note that results are very promising, in two cases

Training set	Recognition Error Rate
1^{st}	0%
2^{nd}	2.22%
3^{rd}	2.22%
4^{th}	2.22%
5^{th}	2.22%
6^{th}	0%

Table 1. Recognition Error Rates on 6 different combinations of training and testing sets.

the system reaches a perfect classification accuracy, and in the others it makes only one error. Clearly only nine subjects for the testing phase are not enough to have statistically significant results: nevertheless, a first impression about the performances of the proposed approach could be derived, giving a promising confidence for future developments. Anyway, a more deep testing, involving more subjects and more environmental changes, will be topic of future investigations.

5 Conclusions

In this paper a new complete low cost system for 3D face recognition has been presented. The 3D face is acquired using a stereo methodology, that does not require any expensive range sensors. The classification step is performed using Support Vector Machines and Multilevel B-Splines coefficients. Preliminary experimental evaluation has produced encouraging results, making the proposed system a promising low cost face recognition system.

References

1. Chellappa, R., Wilson, C., Sirohey, S.: Human and machine recognition of faces: a survey. Proceedings of IEEE **83** (1995) 705–740
2. Ming-Hsuan, Y., Kriegman, D., Ahuja, N.: Detecting faces in images: a survey. IEEE Trans. on Pattern Analysis and Machine Intelligence **24** (2002) 34–58
3. Ho-Chao, H., Ming, O., Wu, J.L.: Automatic feature point extraction on a human face in model-based image coding. Optical Engineering **32** (1993) 1571–1580
4. Lapresté, J., Cartoux, J., Richetin, M.: Face recognition from range data by structural analysis. In: Syntactic and Structural Pattern Recognition, NATO ASI Series (1988) 303–314
5. Gordon, G.: Face Recognition from depth and curvature. PhD thesis, Harvard University (1991)
6. Gordon, G.: Face recognition based on depth and curvature features. In: Proc. of Int. Conf. on Computer Vision and Pattern Recognition. (1992) 808–810
7. Achermann, B., Jiang, X., Bunke, H.: Face recognition using range images. In: Proc. of Int. Conf. on Virtual Systems and MultiMedia. (1997) 129–136
8. Achermann, B.: Face recognition using range images. PhD thesis, Institute of Computer Science and Applied Mathematics, University of Bern (1998)
9. Achermann, B., Bunke, H.: Classifying range images of human faces with hausdorff distance. In: Proc. of Int. Conf. on Pattern Recognition. Volume 2. (2000) 809–813
10. Dhond, U.R., Aggarwal, J.K.: Structure from stereo – a review. IEEE Trans. on System Man and Cybernetics **19** (1989) 1489–1510
11. Lee, S., Wolberg, G., Shin, S.Y.: Scattered data interpolation with multilevel b-splines. IEEE Trans. on Visualization and Computer Graphics **3** (1997) 228–244
12. Burges, C.: A tutorial on support vector machine for pattern recognition. Data Mining and Knowledge Discovery **2** (1998) 121–167
13. Pontil, M., Verri, A.: Support vector machines for 3-d object recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence **20** (1998) 637–646
14. Bicego, M., Iacono, G., Murino, V.: Face recognition with multilevel b-splines and support vector machines. In: Proc. of ACM SIGMM Multimedia Biometrics Methods and Applications Workshop. (2003)
15. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: Proc. of Int. Conf. on Computer Vision, Corfu, Greece (1999)
16. Fusiello, A., Trucco, E., Verri, A.: A compact algorithm for rectification of stereo pairs. Machine Vision and Applications **12** (2000) 16–22
17. Fusiello, A., Castellani, U., Murino, V.: Relaxing symmetric multiple windows stereo using markov random fields. In M.Figureido, Zerubia, J., Jain, A., eds.: Energy Minimization Methods in Computer Vision and Pattern Recognition. Number 2124 in Lecture Notes in Computer Science, Springer (2001) 91–104

18. Trucco, E., Verri, A.: Introductory Techniques for 3-D Computer Vision. Prentice-Hall (1998)
19. Guo, G., Li, S.Z., , Kapluk, C.: Face recognition by support vector machines. Image and Vision Computing **19** (2001) 631–638
20. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag (1995)