3D Face Recognition without Facial Surface Reconstruction

Alexander M. BRONSTEIN, Michael M. BRONSTEIN, Ron KIMMEL and Alon SPIRA

Abstract—**Recently, a 3D face recognition approach based on geometric invariant signatures, has been proposed. The key idea of the algorithm is a representation of the facial surface, invariant to isometric deformations, such as those resulting from facial expressions. One of the crucial stages in the construction of the geometric invariants is the measurement of geodesic distances on triangulated surfaces, carried out by fast marching on triangulated domains (FMTD). Proposed here is a method, which uses only the metric tensor of the surface for geodesic distance computation. When combined with photometric stereo used for facial surface acquisition, it allows constructing a bending-invariant representation of the face without reconstructing the 3D surface.**

***Index Terms*--face recognition, fast marching, photometric stereo, multidimensional scaling.**

I. Introduction

FACE recognition is a biometric method that unlike other biometrics, is non-intrusive and can be used even without the subject's knowledge. State-of-the-art face recognition systems are based on a 40-year heritage of 2D algorithms, dating back to the early 1960s [1]. The first face recognition methods used the geometry of key points (like the eyes, nose and mouth) and their geometric relationships (angles, lengths, ratios, etc.). In 1991, Turk and Pentland applied principal component analysis (PCA) to face imaging [2]. This has become known as the *eigenface* algorithm and is now a golden standard in face recognition. Later, algorithms inspired by eigenfaces that use similar ideas were proposed (see [3], [4], [5]).

However, all the 2D (image-based) face recognition methods appear sensitive to illumination conditions, head orientations, facial expressions and makeup. These limitations of 2D methods stem directly from the limited information about the face contained in a 2D image. Recently, it became evident that the use of 3D data of the face can be of great help as 3D information is viewpoint and lighting-condition independent, i.e. lacks the "intrinsic" weaknesses of 2D approaches.

Gordon showed that combining frontal and profile views can improve recognition accuracy [6]. This idea was extended by Beumier and Acheroy, who compared central and lateral profiles from the 3D facial surface, acquired by a structured light range camera [7]. This approach demonstrated better robustness to head orientations. Another attempt to cope with the problem of head pose was presented by Huang *et al*. using 3D morphable head models [8]. Mavridis *et al*. incorporated a range map of the face into the classical face recognition algorithms based on PCA and hidden Markov models [9]. Particularly, this approach showed robustness to large variations in color, illumination and use of cosmetics, and also allowed separating the face from a cluttered background.

However, none of the approaches proposed heretofore was able to overcome the problems resulting from the non-rigid nature of the human face. For example, Beumier and Acheroy failed to perform accurate global surface matching, and observed that the recognition accuracy decreased when too many profiles were used [7]. The difficulty in performing accurate surface matching of facial surfaces was one of the primary limiting factors of other 3D face recognition algorithms as well.

An attempt to overcome these difficulties has been recently proposed in [10], using the bending invariant canonical forms [11]. In this approach, the facial surface is converted into a representation, which is practically identical for different postures of the face. One of the key stages in the construction of the bending invariant representation is the computation of the geodesic distances between points on a triangulated manifold.

In this work, we present a variation of fast marching on triangulated domains (FMTD), capable of computing geodesic distances given only the metric tensor of the surface. We propose to combine this algorithm with the photometric stereo method for facial surface acquisition. Photometric stereo is a cheap and simple approach, producing the metric tensor without reconstructing the surface. As the result, a simple and fast face recognition method is obtained.

II. Surface Acquisition

The face recognition algorithm discussed in this paper treats faces as three-dimensional surfaces. It is therefore necessary to obtain first the facial surface of the subject that we are trying to recognize. Here, we focus on methods, that produce the surface gradient. As it will be shown in Section IV, the actual surface reconstruction is not needed, saving computational effort and reducing numerical errors.

Alexander (alexbron@ieee.org) and Michael Bronstein (bronstein@ieee.org) are with the Department of Electrical Engineering, Technion. Ron Kimmel (ron@cs.technion.ac.il) and Alon Spira (salon@cs.technion.ac.il) are with the Department of Computer Science, Technion.

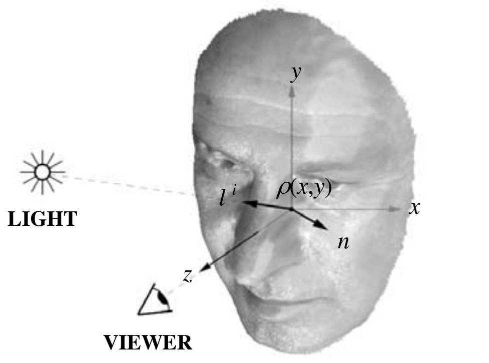

Fig. 1. The photometric stereo acquisition scheme.

A. Photometric stereo

The photometric stereo technique consists of obtaining several pictures of the same subject in different illumination conditions and extracting the 3D geometry by assuming a Lambertian reflection model. We assume that the facial surface, represented as a function, is viewed from a given position along the *z*-axis. The object is illuminated by a source of parallel rays directed along l^i (Figure 1).

We assume a Lambertian reflection model, i.e. the observed image is given by

$$I^i(x,y) = \rho(x,y)\, n(x,y) \cdot l^i, \tag{1}$$

where $\rho(x,y)$ is the object albedo, and $n(x,y)$ is the normal to the object surface, expressed as

$$n(x,y) = \frac{\left(-z_x(x,y), -z_y(x,y), 1\right)}{\sqrt{1+\left\|\nabla z(x,y)\right\|_2^2}}. \tag{2}$$

Using matrix-vector notation, (1) can be rewritten as

$$I(x,y) = \mathrm{L}v, \tag{3}$$

where

$$\mathrm{L} = \begin{pmatrix} l_1^1 & l_2^1 & l_3^1 \\ \vdots & \vdots & \vdots \\ l_1^N & l_2^N & l_3^N \end{pmatrix} \;;\; I(x,y) = \begin{pmatrix} I^1(x,y) \\ \vdots \\ I^N(x,y) \end{pmatrix}, \tag{4}$$

and

$$v_1 = -z_x v_3 \;;\; v_2 = -z_y v_3 \;;\; v_3 = \frac{\rho(x,y)}{\sqrt{1+\left\|\nabla z\right\|_2^2}}. \tag{5}$$

Given at least 3 linearly independent illuminations $\left\{l^i\right\}_{i=1}^N$, and the corresponding observations $\left\{I^i\right\}_{i=1}^N$, one can reconstruct the values of ∇z by pointwise least-squares solution

$$v = \mathrm{L}^\dagger I(x,y), \tag{6}$$

where $\mathrm{L}^\dagger = (\mathrm{L}^\mathrm{T}\mathrm{L})^{-1}\mathrm{L}^\mathrm{T}$ denotes the Moore-Penrose pseudoinverse.

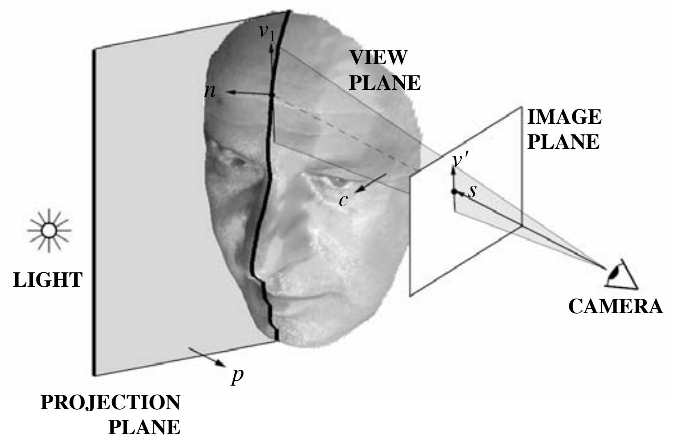

Fig. 2. Structured light acquisition scheme.

When needed, the surface can be reconstructed by solving the Poisson equation

$$\tilde{z}_{xx} + \tilde{z}_{yy} = z_{xx} + z_{yy}, \tag{7}$$

with respect to $\tilde{z}$. In this work, we adopt the photometric stereo approach due to its simplicity.

B. Structured light

Proesmans et al. [12] and Winkelbach and Wahl [13] proposed a shape from 2D edge gradients reconstruction technique, which allows to reconstruct the surface normals (gradients) from two stripe patterns projected onto the object.

The reconstruction technique is based on the fact, that directions of the projected stripes in the captured 2D images depend on the local orientation of the surface in 3D. Classical edge-detecting operators, like Sobel, Canny, etc. can be used to find the direction of the stripe edges.

Figure 2 describes the relation between the surface gradient and the local stripe direction. A pixel in the image plane defines the viewing vector *s*. The stripe direction determines the stripe direction vector v', lying in both the image plane and in the viewing plane. The real tangential vector of the projected stripe v_1 is perpendicular to the normal $c = v' \times s$ of the viewing plane and to the normal *p* of the stripe projection plane. Assuming parallel projection, we obtain

$$v_1 = c \times p. \tag{8}$$

Acquiring a second image of the scene with a rotated stripe illumination relative to the first one, allows to calculate a second tangential vector v_2. Next, the surface normal is computed according to

$$n = v_1 \times v_2. \tag{9}$$

In [14], Winkelbach and Wahl propose to use a single lighting pattern to estimate the surface normal from the local directions and *widths* of the projected stripes.

III. Bending-Invariant Representation

Classical surface matching methods, based on finding an Euclidean transformation of two surfaces which maximizes some shape similarity criterion (see, for example, [15], [16], [17]), are suitable mainly for rigid objects. Human face can by no means be considered as a rigid object since it undergoes deformations resulting from facial expressions. On the other hand, the class of transformations that a facial surface can undergo is not arbitrary, and empirical observations show that facial expressions can be modeled as *isometric* (or length-preserving) transformations. Such transformations do not stretch and not do they tear the surface, or more rigorously, preserve the surface metric. The surfaces resulting from such transformations are called *isometric surfaces*. The requirement from a deformable surface matching algorithm is to find a representation, which is the same for all isometric surfaces.

Schwartz et al. were the first to introduce the use of multidimensional scaling (MDS) as a tool for studying curved surfaces by planar models [18]. Zigelman et al. [19] and Grossman et al. [20] extended some of these ideas to the problem of texture mapping and voxel-based cortex flattening. A generalization of this approach was introduced in the recent work of Elad and Kimmel [11], as a framework for object recognition. They showed an efficient algorithm for constructing a representation of surfaces, invariant under isometric transformations. This method, referred to as bending-invariant canonical forms, is the core of our 3D face recognition framework.

Let us be given a polyhedral approximation of the facial surface, S. One can think of such an approximation as if obtained by sampling the underlying continuous surface on a finite set of points p_i ($i = 1,\ldots,n$), and discretizing the metric δ associated with the surface

$$\delta\left(p_i, p_j\right) = \delta_{ij}. \tag{10}$$

Writing the values of δ_{ij} in matrix form, we obtain the matrix of mutual distances between the surface points. For convenience, we define the *squared* mutual distances,

$$\left(\Delta\right)_{ij} = \delta_{ij}^2. \tag{11}$$

The matrix Δ is invariant under isometric surface deformations, but is not a unique representation of isometric surfaces, since it depends on arbitrary ordering and the selection of the surface points. We would like to obtain a geometric invariant, which would be unique for isometric surfaces on one hand, and will allow using simple rigid surface matching algorithms to compare such invariants on the other. Treating the squared mutual distances as a particular case of *dissimilarities*, one can apply a dimensionality-reduction technique called *multidimensional scaling* (MDS) in order to embed the surface into a low-dimensional Euclidean space $\mathbf{R}^m$. This is equivalent to finding a mapping between two metric spaces,

$$\varphi : \left(\mathrm{S}, \delta\right) \rightarrow \left(\mathrm{R}^m, d\right) \quad ; \quad \varphi\left(p_i\right) = x_i, \tag{12}$$

that minimizes the embedding error,

$$\varepsilon = f\left(\left|\delta_{ij} - d_{ij}\right|\right) \quad ; \quad d_{ij} = \left\|x_i - x_j\right\|_2, \tag{13}$$

for some monotone function *f* that sums over all *ij*.

The obtained *m*-dimensional representation is a set of points $x_i \in \mathbf{R}^m$ ($i = 1,\ldots,n$), corresponding to the surface points p_i. Different MDS methods can be derived using different embedding error criteria [21].

A particular case is the *classical scaling*, introduced by Young and Householder [22]. The embedding in $\mathbf{R}^m$ is performed by double-centering the matrix Δ

$$\mathrm{B} = -\tfrac{1}{2}\mathrm{J}\Delta\mathrm{J} \tag{14}$$

(here $\mathrm{J} = \mathrm{I} - \frac{1}{n}\mathrm{U}$; I is a $n\times n$ identity matrix, and U is a matrix consisting entirely of ones). The first m eigenvectors e_i, corresponding to the m largest eigenvalues of B, are used as the embedding coordinates

$$x_i^j = e_i^j \quad ; \quad i=1,\ldots,n;\ j=1,\ldots,m, \tag{15}$$

where x_i^j denotes the *j*-th coordinate of the vector x_i. We refer to the set of points x_i obtained by the MDS as the *bending-invariant canonical form* of the surface; when m=3, it can be plotted as a surface. Standard rigid surface matching methods can be used in order to compare between two deformable surfaces, using their bending-invariant representations instead of the surfaces themselves. Since the canonical form is computed up to a translation, rotation, and reflection transformation, to allow comparison between canonical forms, they must be *aligned*. This can be done, for instance, by setting the first-order moments (center of mass) and the mixed second-order moments to zero (see [23]).

IV. Measuring Geodesic Distances

One of the crucial steps in the construction of the canonical form of a given surface, is an efficient algorithm for the computation of geodesic distances on surfaces, that is, δ_{ij}. A numerically consistent algorithm for distance computation on triangulated domains, henceforth referred to as *fast marching on triangulated domains* (FMTD), was used by Elad and Kimmel [11]. FMTD was proposed by Kimmel and Sethian [24] as a generalization of the *fast marching method* [25]. Using FMTD, the geodesic distances between a surface vertex and the rest of the n surface vertices can be computed in O(n) operations. Measuring distances on manifolds was also done before for graphs of functions [26] and implicit manifolds [27].

Since the main focus of this paper is how to avoid the surface reconstruction, we present a modified version of FMTD, which computes the geodesic distances on a surface, using the values of the surface gradient ∇z only. These values can be obtained, for example, from photometric stereo or structured light.

The facial surface can be thought of as a parametric manifold, represented by a mapping $\mathrm{X} : \mathbf{R}^2 \rightarrow \mathbf{R}^3$ from the parameterization plane $\mathrm{U} = \left(u^1, u^2\right) = \left(x, y\right)$ to the parametric manifold

$$\mathrm{X}\left(\mathrm{U}\right) = \left(x^1\left(u^1, u^2\right), x^2\left(u^1, u^2\right), x^3\left(u^1, u^2\right)\right); \tag{16}$$

which, in turn, can be written as

$$\mathrm{X}\left(\mathrm{U}\right) = \left(x, y, z\left(x, y\right)\right). \tag{17}$$

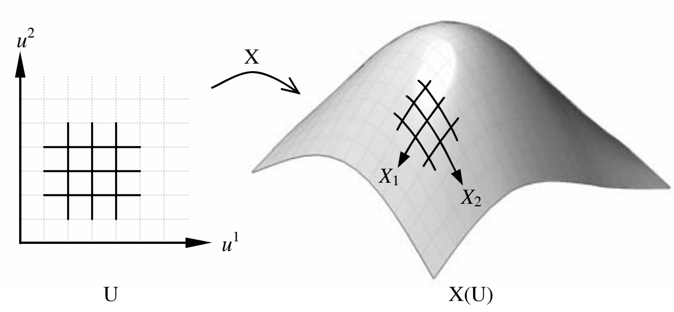

Fig. 3. The orthogonal grid on the parameterization plane U is transformed into a non-orthogonal one on the manifold X(U).

The derivatives of X with respect to u^i are defined as $X_i = \frac{\partial}{\partial u^i}\mathrm{X}$, and they constitute a non-orthogonal coordinate system on the manifold (Figure 3). In the particular case of (17),

$$X_1 = (1, 0, z_x),\; X_2 = (0, 1, z_y)\,. \tag{18}$$

The distance element on the manifold is

$$ds = \sqrt{g_{ij}u^i u^j}\ , \tag{19}$$

where we use Einstein's summation convention, and the metric tensor g_{ij} of the manifold is given by

$$\left(g_{ij}\right) = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} = \begin{pmatrix} X_1 \cdot X_1 & X_1 \cdot X_2 \\ X_2 \cdot X_1 & X_2 \cdot X_2 \end{pmatrix}. \tag{20}$$

The classical fast marching method [25] calculates distances in an orthogonal coordinate system. The numerical stencil for the update of a grid point consists of the vertices of a right triangle. In our case, $g_{12} \neq 0$ and the resulting triangles are not necessarily right ones. If a grid point is updated by a stencil which is an obtuse triangle, a problem may arise. The values of one of the points of the stencil might not be set in time and cannot be used. There is a similar problem with fast marching on triangulated domains which include obtuse triangles [24].

Our solution is similar to that of [24]. We perform a preprocessing stage for the grid, in which we split every obtuse triangle into two acute ones (see Figure 4). The split is performed by adding an additional edge, connecting the updated grid point with a non-neighboring grid point. The distant grid point becomes part of the numerical stencil. The need for splitting is determined according to the angle between the non-orthogonal axes at the grid point. It is calculated by

$$\cos\alpha = \left(\frac{X_1 \cdot X_2}{\|X_1\|\|X_2\|}\right) = \frac{g_{12}}{\sqrt{g_{11}g_{22}}}\,. \tag{21}$$

If $\cos\alpha = 0$, the axes are perpendicular, and no splitting is required. If $\cos\alpha < 0$, the angle α is obtuse and should be split. The denominator of (21) is always positive, so we need only check the sign of the numerator g_{12}. In order to split an angle, we should connect the updated grid point with another point, located *m* grid points from the point in the X_1 direction, and *n* grid points in the X_2 direction (*m* and *n* can be negative).

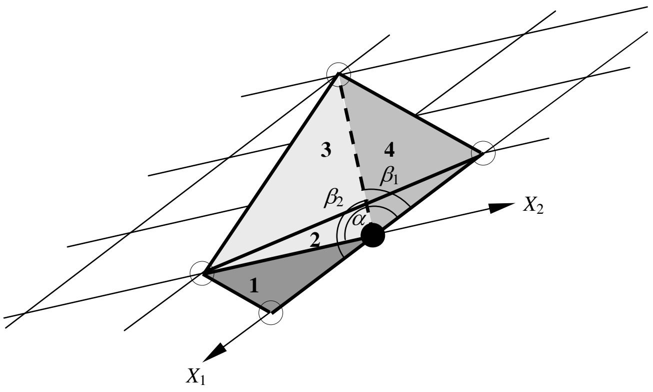

Fig. 4 The numerical support for the non-orthogonal coordinate system. Triangle 1 gives a proper numerical support, yet triangle 2 is obtuse. It is replaced by triangle 3 and triangle 4.

The point is a proper supporting point, if the obtuse angle is split into two acute ones. For $\cos\alpha < 0$ this is the case if

$$\begin{aligned}\cos\beta_1 &= \left(\frac{X_1 \cdot (mX_1 + nX_2)}{\|X_1\|\|mX_1 + nX_2\|}\right) = \\ &= \frac{mg_{11} + ng_{12}}{\sqrt{g_{11}\left(m^2 g_{11} + 2mng_{12} + n^2 g_{22}\right)}} > 0\end{aligned} \tag{22}$$

and

$$\begin{aligned}\cos\beta_2 &= \left(\frac{X_2 \cdot (mX_1 + nX_2)}{\|X_2\|\|mX_1 + nX_2\|}\right) = \\ &= \frac{mg_{12} + ng_{22}}{\sqrt{g_{22}\left(m^2 g_{11} + 2mng_{12} + n^2 g_{22}\right)}} > 0.\end{aligned} \tag{23}$$

Also, here it is enough to check the sign of the numerators. For $\cos\alpha > 0$, $\cos\beta_2$ changes its sign and the constraints are

$$mg_{11} + ng_{12} > 0, \quad mg_{12} + ng_{22} < 0\,. \tag{24}$$

This process is done for all grid points. Once the preprocessing stage is done, we have a suitable numerical stencil for each grid point and we can calculate the distances.

The numerical scheme used is similar to that of [24], with the exception that there is no need to perform the unfolding step. The supporting grid points that split the obtuse angles can be found more efficiently. The required triangle edge lengths and angles are calculated according to the surface metric g_{ij} at the grid point, which, in turn, is computed using the surface gradients z_x, z_y. A more detailed description appears in [28].

V. 3D Face Recognition

As a first step, the 3D face recognition system acquires the surface gradient ∇z, as discussed in Section II. At the second stage, the raw data are preprocessed. Preliminary processing, such as centering and cropping can be carried out by simple pattern matching, which can use the eyes as the most recognizable feature of the human face.

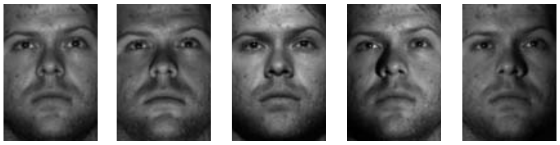
(0°,0°) (0°,-20°) (0°,+20°) (-25°,0°) (+25°,0°)

Fig. 5. Different illuminations of the face. Numbers in brackets indicate the azimuth and the elevation angle, respectively, determining the illumination direction.

The facial contour should also be extracted in order to limit the processing to the surface belonging only to the face itself. Preprocessing should emphasize those sections of the face less susceptible to alteration and exclude the parts that can be easily changed (e.g. hair). In this work, the preprocessing stage was limited to cropping the triangulated manifold by removing the parts lying outside an ellipse (in the geodesic sense) centered at the nose tip.

Next, an $n \times n$ matrix of geodesic distances is created by applying FMTD from each of the n selected vertices. Then, MDS is applied to the distance matrix, producing a canonical form of the face in a low-dimensional Euclidean space (three-dimensional in all our experiments).

The canonical form obtained this way is compared with a database of templates corresponding to other subjects (one-to-many match), or to a single stored template (one-to-one match). If the correspondence of the compared canonical form falls within a certain statistical range of values, the match is considered to be valid.

Finding correspondence between the canonical forms is a problem by itself and brings us back to the surface matching problem. However, the major difference here is the fact that the matching is performed between the rigid canonical forms carrying the intrinsic object geometry, rather than the deformable facial surfaces themselves. Thus even the simplest rigid surface matching algorithms produce plausible results.

In this work, we adopted the moments [23] due to its simplicity. The canonical form's (p, q, r)-th moment is given by

$$M_{pqr} = \sum_n \left(x_n^1\right)^p \left(x_n^2\right)^q \left(x_n^3\right)^r \qquad (25)$$

where x_n^i denotes the i^{th} coordinate of the n^{th} point in the canonical surface samples. In order to compare between two canonical forms, the vector $\left(M_{p_1 q_1 r_1}, \ldots, M_{p_M q_M r_M}\right)$, termed as the *moments signature*, is computed for each surface. The Euclidean distance between two moments signatures measures the dissimilarity between the two surfaces.

VI. Experimental Results

We performed an experiment, which demonstrates that comparison of canonical forms obtained without actual facial surface reconstruction is better than reconstruction and direct comparison of the surfaces. The Yale Face Database B [29] was used. The database consisted of high-resolution grayscale images of different instances of 10 subjects of both Caucasian and Asian type, taken in controlled illumination conditions (Figure 5). Some instances of 7 subjects were taken from the database for the experiment.

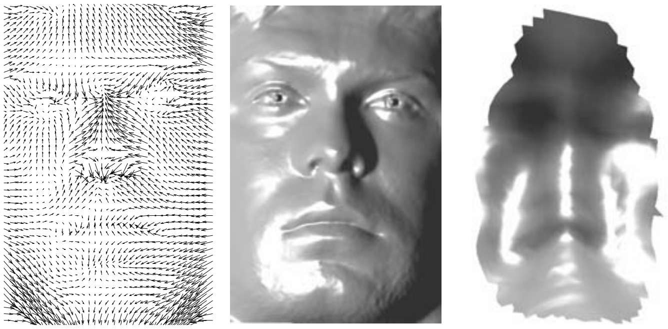

Fig. 6. Surface gradient field (left), reconstructed surface (center) and its bending-invariant canonical form represented as a surface (right).

Direct surface matching consisted of the retrieval of the surface gradient according to (6) using 5 different illumination directions, reconstruction of the surface according to (7), alignment and computation of the surface moments signature according to (25). Canonical forms were computed from the surface gradient, aligned and converted into a moment signature according to (25).

In order to get some impression of the algorithms accuracy, we converted the relative distances between the subjects produced by each algorithm into 3D proximity patterns (Figure 7). These patterns, representing each subject as a point in $\mathbf{R}^3$, were obtained by applying MDS to the relative distances (with a distortion less than 1%).

The entire cloud of dots was partitioned into clusters formed by instances of the subjects C_1—C_7. Visually, the more C_i are compact and distant from other clusters, the more accurate is the algorithm. Quantitatively, we measured (i) the variance σ_i of C_i and (ii) the distance d_i between the centroid of C_i and the centroid of the nearest cluster.

Table I shows a quantitative comparison of the algorithms. Inter-cluster distances d_i are given in units of the variance σ_i. Clusters C_5—C_7, consisting of a single instance of the subject are not presented in the table. The use of canonical forms improved the cluster variance and the inter-cluster distance by about one order of magnitude, compared to direct facial surface matching.

TABLE I
Face Recognition Accuracy

CLUSTER	σ_{DIRECT}	d_{DIRECT}	$\sigma_{\text{CANONICAL}}$	$d_{\text{CANONICAL}}$
C_1	0.1749	0.1704	0.0140	4.3714
C_2	0.2828	0.3745	0.0120	5.1000
C_3	0.0695	0.8676	0.0269	2.3569
C_4	0.0764	0.7814	0.0139	4.5611

VII. Conclusions

We have shown how to perform 3D face recognition according to [10], without reconstructing the facial surface.

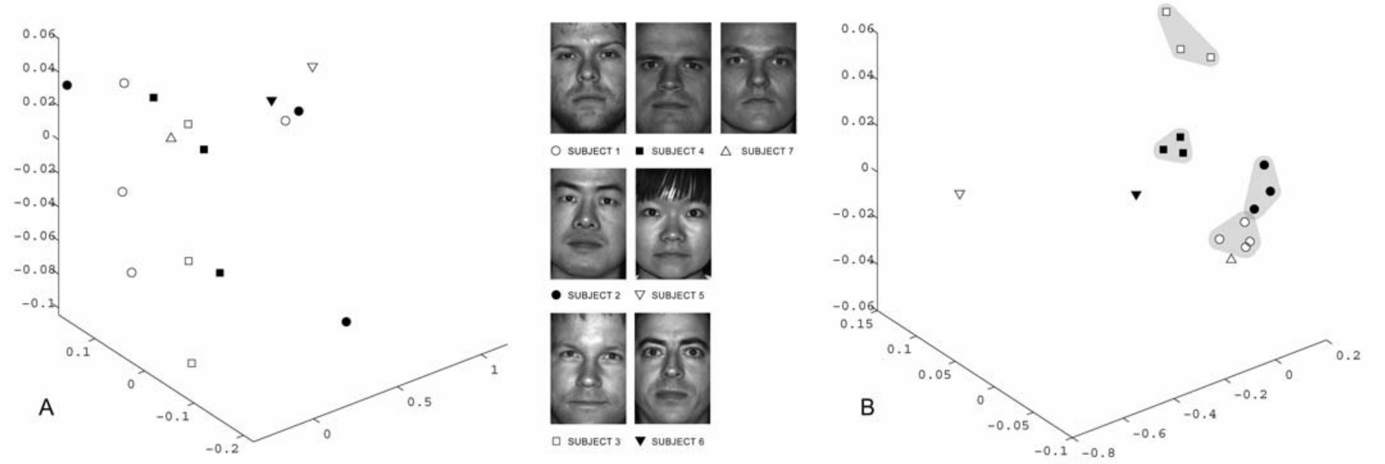

Fig. 7. Visualization of the face recognition results as three-dimensional proximity patterns. Subjects from the face database represented as points obtained by applying MDS to the relative distances between subjects. Shown here: straightforward surface matching (A) and canonical forms (B).

3D face recognition based on bending-invariant representations, unlike previously proposed solutions, makes face recognition robust to facial expressions, head orientations and illumination conditions. Our approach shown here allows an efficient use of simple 3D acquisition techniques (e.g. photometric stereo) for fast and accurate face recognition. Experimental results demonstrate superiority of our approach over straightforward rigid surface matching.

REFERENCES

[1] W. W. Bledsoe, The model method in facial recognition. *Technical report PRI 15, Panoramic Research Inc., Palo Alto*, 1966.

[2] M. Turk and A.. Pentland,, Face recognition using eigenfaces, *Proc. CVPR*, pp. 586-591, 1991.

[3] V. I. Belhumeur, J. P. Hespanha and D. J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, *IEEE Trans. PAMI* 19(7), pp. 711-720, 1997.

[4] B. J. Frey, A. Colmenarez and T. S. Huang, Mixtures of local linear subspaces for face recognition, *Proc. CVPR*, pp. 32-37, 1998.

[5] B. Moghaddam, T. Jebara and A. Pentland, Bayesian face recognition. *Technical Report* TR2000-42, Mitsubishi Electric Research Laboratories, 2000.

[6] G. Gordon, Face recognition from frontal and profile views, *Proc. Int'l Workshop on Face and Gesture Recognition*, pp. 47-52, 1996.

[7] C. Beumier and M. P. Acheroy, Automatic face authentication from 3D surface*, Proc. British Machine Vision Conf. (BMVC)*, pp 449-458, 1998.

[8] J. Huang, V. Blanz and B. Heisele, Face recognition using component-based SVM classification and morphable models, *SVM* 2002, pp. 334-341, 2002.

[9] N. Mavridis, F. Tsalakanidou, D. Pantazis, S. Malassiotis and M. G. Strintzis, The HISCORE face recognition application: Affordable desktop face recognition based on a novel 3D camera, *Proc. Int'l Conf. on Augmented Virtual Environments and 3D Imaging (ICAV3D)*, Mykonos, Greece, 2001.

[10] Anonymous authors, 3D face recognition using geometric invariants, Submitted for publication 2003.

[11] A. Elad, R. Kimmel, Bending invariant representations for surfaces, *Proc. CVPR*, pp. 168-174, 2001.

[12] M. Proesmans, L. Van Gool and A. Oosterlinck, One-shot active shape acquisition, *Proc. Internat. Conf. Pattern Recognition*, Vienna, Vol. C, pp. 336- 340, 1996.

[13] S. Winkelbach and F. M. Wahl, Shape from 2D edge gradients, *Pattern Recognition, Lecture Notes in Computer Sciences* 2191, pp. 377-384, Springer, 2001.

[14] S. Winkelbach and F. M. Wahl, Shape from single stripe pattern illumination, L. Van Gool (Ed): *Pattern Recognition (DAGM 2002)*, *Lecture Notes in Computer Science* 2449, pp. 240-247, Springer, 2002.

[15] O. D. Faugeras and M. Hebert, A 3D recognition and positioning algorithm using geometrical matching between primitive surfaces, *Proc. 7th Int'l Joint Conf. on Artificial Intelligence*, pp. 996–1002, 1983.

[16] P. J. Besl, The free form matching problem. *Machine vision for three-dimensional scene*, In: Freeman, H. (ed.) New York Academic , 1990.

[17] G. Barequet and M. Sharir, Recovering the position and orientation of free-form objects from image contours using 3D distance map*, IEEE Trans. PAMI*, 19(9), pp. 929–948, 1997.

[18] E. L. Schwartz, A. Shaw and E. Wolfson, A numerical solution to the generalized mapmaker's problem: flattening nonconvex polyhedral surfaces, *IEEE Trans. PAMI*, 11, pp. 1005-1008, 1989.

[19] G. Zigelman, R. Kimmel and N. Kiryati, Texture mapping using surface flattening via multi-dimensional scaling, *IEEE Trans. Visualization and Comp. Graphics*, 8, pp. 198-207, 2002.

[20] R. Grossman, N. Kiryati and R. Kimmel, Computational surface flattening: a voxel-based approach, *IEEE Trans. PAMI*, 24, pp. 433-441, 2002.

[21] I. Borg and P. Groenen, *Modern multidimensional scaling - theory and applications*, Springer, 1997.

[22] G. Young and G. S. Householder, Discussion of a set of points in terms of their mutual distances, *Psychometrika* 3, pp. 19-22, 1938.

[23] A. Tal, M. Elad and S. Ar, Content based retrieval of VRML objects – an iterative and interactive approach", *EG Multimedia*, 97, pp. 97-108, 2001.

[24] R. Kimmel and J. A. Sethian, Computing geodesic on manifolds. *Proc. US National Academy of Science* 95, pp. 8431–8435, 1998.

[25] J. A. Sethian, A review of the theory, algorithms, and applications of level set method for propagating surfaces. *Acta numerica*, pp. 309-395, 1996.

[26] J. Sethian and A. Vladimirsky, Ordered upwind methods for static Hamilton-Jacobi equations: theory and applications*.* *Technical Report PAM* 792 (UCB), Center for Pure and Applied Mathematics, 2001 Submitted for publication to SIAM Journal on Num. Anal., 2001.

[27] F. Mémoli and G. Sapiro, Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. *Journal of Computational Physics*, 173(2), pp. 730-764, 2001.

[28] Anonymous authors, An efficient solution to the eikonal equation on parametric manifolds. Submitted for publication, 2003.

[29] Yale Face Database B. Available: http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html