

3D Face Recognition in Continuous Spaces

Francisco José Silva Mata¹, Elaine Grenot Castellanos¹,
Alfredo Muñoz-Briseño¹, Isneri Talavera-Bustamante¹,
and Stefano Berretti²(✉)

¹ CENATAV, Havana, Cuba

² University of Florence, Florence, Italy
stefano.berretti@unifi.it

Abstract. This work introduces a new approach for face recognition based on 3D scans. The main idea of the proposed method is that of converting the 3D face scans into a functional representation, performing all the subsequent processing in the continuous space. To this end, a model alignment problem is first solved by combining graph matching and clustering. Fiducial points of the face are initially detected by analysis of continuous functions computed on the surface. Then, the alignment is performed by transforming the geometric graphs whose nodes are the critical points of the representative function of the surface in previously determined subspaces. A clustering step is finally applied to correct small displacement in the models. The 3D face representation is then obtained on the aligned models by functions carefully selected according to mathematical and computational criteria. In particular, the face is divided into regions, which are treated as independent domains where a set of functions is determined by fitting the surface data using the least squares method. Experimental results demonstrate the feasibility of the method.

Keywords: 3D face recognition · Functional representation

1 Introduction

Models of the face acquired by 3D devices consist of dense point clouds, where points correspond to coordinates of the face surface discretely sampled by the capture device. For high resolution 3D scans, a very large number of points is typically used to represent the face, and triangular mesh representations are then derived to connect points in a structured way. However, this low level representation cannot be used directly to compare faces in recognition tasks, but appropriate descriptors that reduce the high dimensionality of points keeping, at the same time, salient features of the face should be derived.

Face recognition using either high resolution or low-resolution 3D scans has received an increasing interest in the last few years (for a thorough discussion of existing methods we refer to the survey in [7] and the literature review in [3, 18]). In general, 3D face recognition approaches proposed in the literature can be grouped as *global* (or *holistic*), and *local* (or *region-based*). Hybrid approaches

that combine solutions in these two categories are also possible as well as multimodal approaches that combine together 2D and 3D methods. Among the aspects that still are critical for most state of the art methods, we can count recognition across scans with different resolutions (high- or low-resolution as for consumer cameras like Kinect [5]), and recognition of scans with large/extreme pose variations or occlusions, which requires partial face matching. This is also reflected by the few face datasets that include face scans with different resolutions [5] or partial acquisitions [1, 2, 17]. Global 3D face representations for partial face matching have been proposed in a limited number of works [8, 14]. More successful and scalable solutions used local representations of the face. In fact, one possible way to solve the problem of missing data in 3D faces is to detect locally the absence of regions of the face and use the existing data to reconstruct the missing parts (for example, exploiting the hypothesis of face symmetry to recover missing data in the case of scans with large pose variations [15]). The reconstructed scans can then be used as input to conventional 3D face recognition methods [10]. Tackling the problem from an opposite perspective, some methods divided the face into regions and tried to restrict the match to uncorrupted parts of the face [11, 12]. Most of these methods used landmarks of the face to identify the regions to be matched; however, facial landmarks are difficult to detect when the pose significantly deviates from the frontal one. In addition, since parts of the regions can be missing or occluded, the extraction of effective descriptors is hindered so that regions comparison is mostly performed using rigid (ICP) or elastic registration (deformable models). Approaches that use keypoints of the face solve some of these limitations. Rather than relying on the detection of specific regions of the face that can fail in the presence of occlusions and missing parts, they detect keypoints on the face surface and describe the face locally at the keypoints. Matching keypoints can thus naturally account for occlusions and missing parts of the face [4, 13].

In this work, we propose an original solution to 3D face recognition that, on the one hand, exploits keypoints for face alignment, on the other, accurately represents locally the face surface. Our solution is robust to the presence of scans acquired with large pose variations (and thus with missing parts), and is based on two main original contributions: a graph-based solution to align 3D face scans with missing parts; a *functional representation* that provides a locally continuous approximation of the face surface. The idea of approximating the face surface with continuous functions is a well known and used techniques in Computer Graphics. In that case, recovering the exact form of the surface is important for visualization. Differently, in the case of recognition tasks, the necessary optimization of the functional model must be able to obtain more discriminative representations of the face with the least number of coefficients. Indeed, the process of optimizing the functional model as well as the selection of the set of base functions are crucial for this method [16]. Functional representations are attractive for the recognition scenario because they show some interesting aspects. First, they demonstrate great power in compacting the data thanks to the small-dimensional vectors of used coefficients. In addition, they

allow recovering the original continuous nature of biometric objects or their parts. This representation also allows capturing the correlation between the different values of 2D pixels or 3D vertices. The ability to use the existing theory of continuous functions often simplifies calculations and analysis. The representation of dynamic aspects of the original data and the possibility of extracting some important features through the analysis of the properties of functions, such as monotonicity, derivability and smoothness, makes attractive the use of functions to represent data that naturally vary in space continuously. However, an essential element to make functional representations comparable is that the origin of coordinates and the directions of the axes coincide across different objects. To achieve this, a process of prior alignment of the 3D faces is necessary. To this end, in this paper, we also propose a solution for aligning face scans with missing parts. This relies on three steps: first, the face is divided in rectangular domains and fiducial points of the face are detected as critical points of a local functional representation of the face surface based on *Local Thin Plate Bivariate Splines* (LTPBVS) [6]; then, a graph-like structure is constructed from the fiducial points connections; finally, matching these graphs permits face alignment.

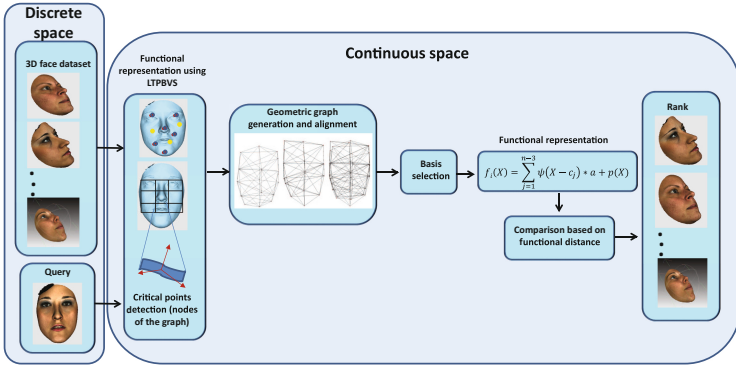


Fig. 1. The proposed 3D face recognition approach in continuous space

The processing steps of the proposed solution are summarized in Fig. 1: first, face scans are subdivided, approximated with LTPBVS and aligned using a graph of critical points; then, a LTPBVS basis is selected to approximate the face surface; finally, coefficients of the functional representation are used in the match. The rest of the paper is organized as follows: in Sect. 2, the method used for the detection of critical points of 3D faces is presented; the construction of a graph based on these points and its use for face alignment are illustrated in Sect. 3; the functional representation of the face is discussed in Sect. 4; experiments performed to evaluate our proposed method are reported in Sect. 5; discussion and conclusions in Sect. 6 close the paper.

2 Detection of Characteristic Points in 3D Faces

For the detection of characteristic points in 3D faces, Principal Component Analysis (PCA) was first performed to normalize the cloud of points of the whole face in such a way that their coordinate axes coincide with the principal components. To this end, given a set of points represented in the matrix A of size $N \times 3$, where each row is a point in space, it was necessary to calculate the covariance matrix. The eigenvectors of this matrix were used as the new coordinate axes: the z -axis captures the direction of the data with the smallest variance, i.e., the eigenvector corresponding to the lowest eigenvalue (this axis is also an estimate of the actual normal vector of the face surface corresponding to the points cloud); the y -axis corresponds to the vector of greatest variance and, finally, the x -axis corresponds to the vector associated with the eigenvalue of intermediate value.

The surface of the face is divided in rectangular domains, and a non-polynomial function is fitted to the surface of each domain. These rectangles have the same size determined according to the mesh size, and are represented as:

$$D_{ij} = \{(x, y, z) : (x, y) \in [x_i, x_i + d] \times [y_j, y_j + t]\}, \quad (1)$$

with $i, j = 1, 2, 3$, and where x_1 and y_1 are the minima of the column vectors X and Y of A , respectively; the other values of x_i and y_i are, respectively, $x_2 = x_1 + d$, $x_3 = x_2 + d$, and $y_2 = y_1 + t$, $y_3 = y_2 + t$. Values of d and t are obtained as:

$$\begin{pmatrix} d \\ t \end{pmatrix} = \frac{1}{3} \begin{pmatrix} x_{max} - x_1 \\ y_{max} - y_1 \end{pmatrix}, \quad (2)$$

where x_{max} and y_{max} are the respective values of X and Y .

The surface that approximates the point cloud in each region (sub-domain) is obtained by a non-polynomial function. To this end, first the centroid of each region D_{ij} is considered as the origin of the local coordinate system, and the coordinate axes are calculated as the local eigenvectors of each sub-domain. The smallest of the three eigenvectors corresponds to the normal direction of each sub-domain. This ensures that the local z -axis is perpendicular to the surface. Then, the function that approximates the region of the points cloud has the form of a scattered translates, namely, a Bivariate Thin-Plate Spline. It uses arbitrary or scattered translates $\psi(\cdot - c_j)$ of one fixed function ψ , in addition to some polynomial terms. Explicitly, such a form describes a function:

$$f(X) = \sum_{j=1}^{n-3} \psi(X - c_j) a_j + p(X), \text{ where } X = (x, y), \quad (3)$$

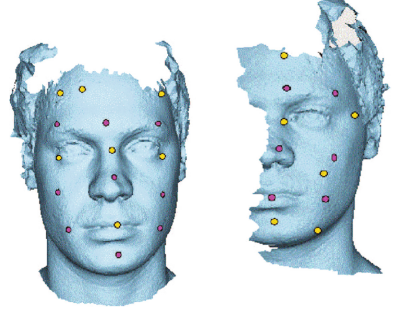
where the basis function is $\psi(X) = \varphi(\|X\|^2)$, with $\|\cdot\|$ the Euclidean norm, and $\varphi(t) = t \log t$; c_j , a sequence of sites called *centers*, and a_j a corresponding sequence of n coefficients with the final three coefficients involved in the polynomial part:

$$P(X) = a_{n-2} \cdot x + a_{n-1} \cdot y + a_n. \quad (4)$$

```

1: Procedure classifyPoints()
2: for  $j = 1$  to 8 do
3:   if  $(x_j, y_j)$  isReal then
4:     if  $\det(h_j) > 0$  then
5:       if  $h_j[0, 0] > 0$  then
6:          $(x_i, y_i) \rightarrow \text{minimum}$ 
7:       else if  $h_j[0, 0] < 0$  then
8:          $(x_i, y_i) \rightarrow \text{maximum}$ 
9:       end if
10:    else if  $\det(h_j) < 0$  then
11:       $(x_i, y_i) \rightarrow \text{saddle}$ 
12:    end if
13:  end if
14: end for
15: return

```



(a) Front capture (b) Side capture

Fig. 2. Procedure *classifyPoints*()**Fig. 3.** Results of the detection process.

The critical points (*maxima*, *minima*, and *saddles*) of the polynomial P correspond to the characteristic points of the face. These points are found with a subsequent inverse transformation to reach the points of the original face. To this end, the *gradient* G of the polynomial P is computed:

$$G(P) = \left(\frac{\partial P}{\partial x}(x, y), \frac{\partial P}{\partial y}(x, y) \right). \quad (5)$$

solving the following system:

$$\begin{cases} \frac{\partial P}{\partial x}(x, y) = 0 \\ \frac{\partial P}{\partial y}(x, y) = 0 \end{cases}. \quad (6)$$

As result, the eight possible solutions $\{(x_j, y_j)\}_{j=1}^8$ for the system are found. An evaluation of every real solution is performed on the *Hessian* matrix H of P . This evaluation is denoted as $h_j = H(P)(x_j, y_j)$. In this way, each real solution is classified according to its type (*minimum*, *maximum* or *saddle*) by following the procedure described in Fig. 2. As can be seen in this procedure, the classification is performed by computing the determinant of h_i and evaluating its first element (Fig. 3 shows some detected critical points).

On the other hand, when the determinant of h turns out to be zero, the point (x_i, y_i) in the polynomial function is evaluated, and its behavior is analyzed in such a way that: if $P(x_i, y_i) < P(x, y)$ it is a maximum; if $P(x_i, y_i) > P(x, y)$ it is a minimum; and if $P(x, y)_{(x,y) < (x_i, y_i)} < P(x_i, y_i) < P(x, y)_{(x,y) > (x_i, y_i)}$ it is a saddle point.

Some automatic adjustments of the position of windows or sub-domains were made to achieve greater efficacy of the method. These adjustments were executed starting by placing the first sub-domain in the approximate area of the nose (usually located in the center of the face for D_{22}), where in almost 100% of the cases there is a detectable maximum. Given the windows:

$$V_{i2} = \{(x, y, z) : (x, y) \in [x_i, x_i + d] \times [y_2, y_2 + t]\}, \quad (7)$$

with $i = 2, 3, \dots$, making shifts of five units to the right $x_2 = x_1 + 5$, $x_3 = x_2 + 5, \dots$, and to the left $x_i = x_1 - 5$, $x_{i+1} = x_2 - 5, \dots$. Until to lose the maximum in both directions and find points of minima and saddles at the end of the nose; it is obtained an intermediate sub-domain that is used like reference for the rest of the windows of the face. The length d of this intermediate window in the x -axis is given by $\frac{x_1^d + x_2^d}{2} - \frac{x_1^i + x_2^i}{2}$, where $x_{1,2}^d$ and $x_{1,2}^i$ are the respective lower and upper boundaries of the final windows given the right and left shifts. Then, being $x_1 = \frac{x_1^i + x_2^i}{2}$, $x_2 = \frac{x_1^i + x_2^i}{2}$ and $x_3 = \frac{x_1^d + x_2^d}{2} - d$, and the remaining windows would be as follows:

$$V_{ij} = \{(x, y, z) : (x, y) \in [x_i, x_i + d] \times [y_j, y_j + t]\}. \quad (8)$$

3 Alignment of Two Faces

Before performing the recognition step between two faces, an alignment must be performed. Let $P_1 = \{p_1, p_2, \dots, p_n\}$ and $P_2 = \{p_1, p_2, \dots, p_n\}$ be the sets of fiducial points extracted from the representations of two 3D faces. Each point of these sets can be represented by the tuple $p_i = (x_i, y_i, z_i, l_i)$, where x_i , y_i and z_i are the coordinates of the described point in \mathbb{R}^3 , and l_i is a label that can take three values depending on the kind of fiducial point detected (i.e., maximum, minimum or saddle).

The proposed alignment is based on finding a labeled geometric graph for each set of points. This is performed by computing Delaunay triangulation in 3D of the sets P_1 and P_2 , denoted by $DT_3(P_1)$ and $DT_3(P_2)$. This triangulation is a generalization of the classic Delaunay triangulation in which no point in P_i is inside the circum-hypersphere of any simplex (tetrahedron) in $DT_3(P_i)$. It is known that $DT_3(P_i)$ is unique if P_i is a set of points in general position. This means that the affine hull of P_i is 3-dimensional and no set of 5 points in P_i lie on the boundary of a ball whose interior does not intersect P_i [9]. In this way, $DT_3(P_i)$ can be decomposed in simplexes, each one conformed by four facets. The main objective of computing $DT_3(P_1)$ and $DT_3(P_2)$ is to find a tolerant to distortions and unique geometrical structure for each P_i .

On the other hand, a labeled geometric graph can be defined as follows:

Definition 1 (Geometric graph). *A geometric graph is a 4-tuple, $G = (V, E, I, K)$, where V is a set of vertexes, $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ is a set of edges (the edge $\{u, v\}$ connects the vertexes u and v), $I : V \rightarrow L_V$ is a function that assigns labels to vertexes where L is the domain of labels and, finally, $K : V \rightarrow \mathbb{R}^3$ is a function that assigns coordinates to vertexes, \mathbb{R} represents the set of real numbers, and $K(u) \neq K(v)$ for each $u \neq v$.*

Using the previous definition and the triangulations $DT_3(P_1)$ and $DT_3(P_2)$, the labeled geometric graphs $G_1 = (V_1, E_1, I_1, K_1)$ and $G_2 = (V_2, E_2, I_2, K_2)$ are obtained, respectively, from P_1 and P_2 . It results that:

- V_i represents the points of P_i ;
- E_i contains all the edges generated by $DT_3(P_i)$;

```

1: Procedure alignPoints( $P_1, P_2$ )
2:  $G_1(V_1, E_1, I_1, K_1) \leftarrow \text{buildGraph}(P_1)$ 
3:  $G_2(V_2, E_2, I_2, K_2) \leftarrow \text{buildGraph}(P_2)$ 
4:  $H \leftarrow \{\emptyset\}$ 
5: for all face  $v_i, v_j, v_k \in G_1$  do
6:   for all face  $v_l, v_m, v_n \in G_2$  do
7:     sortVertexes( $v_i, v_j, v_k$ )
8:     sortVertexes( $v_l, v_m, v_n$ )
9:     if  $I_1(v_i) == I_2(v_l)$  and  $I_1(v_j) == I_2(v_m)$  and  $I_1(v_k) == I_2(v_n)$  then
10:      addTrans( $H, E_1(v_i, v_j), E_2(v_l, v_m)$ )
11:      addTrans( $H, E_1(v_j, v_k), E_2(v_m, v_n)$ )
12:      addTrans( $H, E_1(v_k, v_i), E_2(v_n, v_l)$ )
13:     end if
14:   end for
15: end for
16:  $T_M \leftarrow \text{getMaxRepTrans}(H)$ 
17: applyTransformation( $P_2, T_M$ )
18: return

```

Fig. 4. Procedure *alignPoints*(P_1, P_2)

- I_i is a function that assigns labels from $L_V = \{1, 2, 3\}$ depending on the type of the point represented (i.e., maximum, minimum or saddle);
- K assigns coordinates to the vertexes.

In Fig. 6 a geometric graph generated with this procedure is shown. After this step, a graph matching technique between G_1 and G_2 is done. With this technique, the geometric transformation T that best aligns G_1 with G_2 is found.

In Figs. 4 and 5 the procedures used for aligning two sets of points P_1 and P_2 are shown. In lines 2–3 of the procedure *alignPoints*(), the graphs are created. Then, a map structure H is initialized. In lines 5–15, the faces of each simplex of G_1 are compared with those contained in G_2 . For this, the vertexes of the faces are sorted according to the lengths of its segments. After this, if the analyzed pair of faces have the same labels, the procedure *addTrans*() is called. In this latter procedure, the transformation matrix T used to convert the second segment into the first one is computed. Then, if T or a similar transformation is contained in H , the respective counter is augmented; otherwise a new entry is added with the counter set to 0. Finally, the transformation T_M with higher counter in H is used to rotate and translate P_2 with respect to P_1 .

The main idea of this algorithm is based on finding the geometric transformation T_M that aligns the highest number of edges belonging to G_1 and G_2 . This algorithm assumes that the fiducial points extracted from all the 3D faces have a similar geometric disposition and labeling. As an example, Fig. 6 shows the representation as geometric graphs and the alignment of two faces.

In order to refine the geometric graphs alignment, a posterior clustering procedure is performed. First of all, the PCA algorithm is applied to the whole model to determine the z -axis, as the direction of lower variance. Then, a

```

1: Procedure addTrans( $H, E_1(v_i, v_j), E_2(v_l, v_m)$ )
2:  $s_1 \leftarrow \text{sizeofSegment}(E_1(v_i, v_j))$ 
3:  $s_2 \leftarrow \text{sizeofSegment}(E_2(v_l, v_m))$ 
4: if  $\text{abs}(s_1 - s_2)$  then
5:    $T \leftarrow \text{getTransMatrix}(E_1(v_i, v_j), E_2(v_l, v_m))$ 
6:   if  $\text{isContained}(H, T)$  then
7:      $H(T).\text{counter}++$ 
8:   else
9:      $H \leftarrow \{T, 1\}$ 
10:  end if
11: end if
12: return

```

Fig. 5. Procedure $\text{addTrans}(H, E_1(v_i, v_j), E_2(v_l, v_m))$

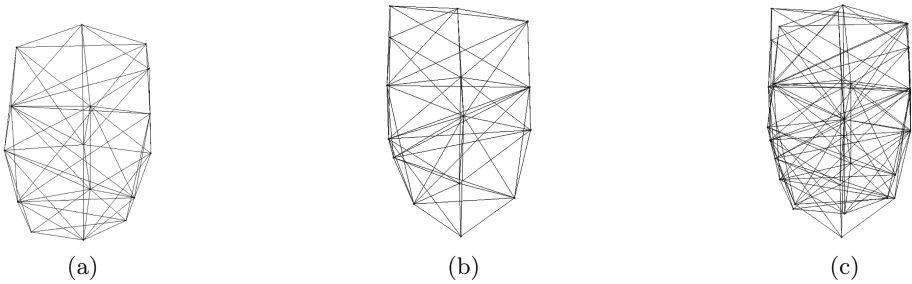


Fig. 6. (a)–(b) Two graphs of faces; (c) alignment of the graphs in (a) and (b)

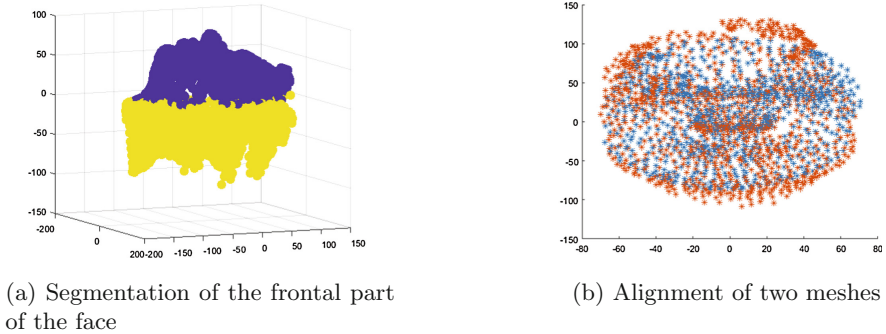


Fig. 7. Refinement of the alignment process

k -means clustering is applied to the z values, in order to segment the frontal part of the face (see Fig. 7a). Finally, PCA is applied again, using as origin the maximum value of the z coordinates found. In this way, the y -axis is given by the direction of lower variance. In Fig. 7 the results of this process are shown.

The main advantages of the proposed method over other state of the art approaches [16] are the following:

- The localization of specific fiducial points in the faces, like pronasal points, are not needed.
- The use of a point set registration algorithm, like ICP is avoided. These algorithms are computationally expensive.
- Our proposal performs the alignment process with high precision, between lateral and frontal views of the faces. This is not possible in previous works.

4 Functional Representation

Once aligned, the next step is to obtain the representation of the points cloud as a surface corresponding to a function $z = f(x, y)$ over a spatial domain. The appropriate domain in terms of its dimensions and geometry must correspond to the completion of the functional representation. As base functions, we use the LTPBVS (see (3)), but adjusted to the surface of the new regions obtained after the alignment of the faces. In this way, the representation is constructed by the same procedure used to detect the points for alignment, which simplifies the implementation of the process. The decision to use LTPBVS is supported by the well known advantages of these functions. Among them, we can mention that LTPBVS produces smooth surfaces, which are infinitely differentiable. Also, they do not have free parameters that need manual tuning.

The matching step is performed by comparing the coefficients of the corresponding representative functions of the faces, in a way similar to [16]. However, in this work we obtain one functional representation for each one of the m regions in which the face is divided. Given two faces F and G , their distance can be computed as in (9), where f_i and g_i are the corresponding functions of the i -th region, defined on a common domain $[a, b] \times [c, d]$ for the norm L_n :

$$d(F, G) = \sum_{i=1}^m \sqrt[n]{\int_a^b \int_c^d |f_i(x, y) - g_i(x, y)|^n dx dy}. \quad (9)$$

5 Experimental Results

The proposed 3D face recognition approach has been evaluated on the 2D/3D Florence dataset [2]. This dataset includes 3D faces acquired with different devices and challenges (i.e., non-frontal pose, presence of hair, neck, shoulders). For the whole dataset, the representations were constructed based on local thin plate bivariate splines (LTPBVS) defined, respectively, on twelve disjoint regions oriented by the normals to the origin and over the correspondent control grid. Thus, in the case of side faces, they contain six disjoint rectangular regions. The face recognition problem was modeled as a classification task, using a k-NN classifier with Euclidean distance. Results are reported in Table 1. For each one of the disjoint regions found, 39 coefficients were computed.

Table 1. Rank-1 recognition accuracy on the 2D/3D Florence face dataset

Method	# coef.	Frontal	Left	Right
Our proposal	468	98%	97.1%	97.2%
GBVS [16]	4096	100%	96.2%	96.4%
LBVS [16]	4608	100%	96.3%	96.7%

It can be noted, the results do not outperform those obtained in a previous approach on the frontal case, but it reports better results on lateral cases. This feature makes the proposal of this work more suitable for environments in which occlusions are common. Also, the number of coefficients used on this approach is lesser than previous works, which reduces the dimension of the data and improves the efficiency of the method.

6 Discussion and Conclusions

Recognizing faces from 3D scans is becoming a problem of increasing interest, with applications in several practical contexts. Though effective solutions exist for the cooperative case, where faces are acquired in frontal pose, the recognition is much more difficult when acquisitions include facial expressions or pose variations (missing parts).

In this paper, we have presented an original 3D face recognition solution, which is capable of recognizing faces also in the case of expressions and missing parts. The proposed method relies on the idea of constructing a functional representation of the face locally. First, keypoints of the face are detected using surface analysis, and they are used to partition the face into local rectangular domains, which are subsequently aligned. Then, the surface is approximated locally to each domain using Local Thin Plate Bivariate Splines (LTPBVS). The LTPBVS provide a descriptive and compact representation of the face, where coefficients of the functions are used for effective and efficient face matching. On the other hand, the proposed alignment method is very robust in presence of position variation or omission of fiducial points. This occurs because the alignment can be performed by using only a small subset of fiducial points, which allows a higher degree of tolerance. The proposed method has good performance even when a certain amount of spurious fiducial points are located. Recognition results obtained on the UF-3D [2] database show performances, which are comparable or superior to state of the art solutions.

References

1. University of Notre Dame biometrics database (2008). <http://www.nd.edu/~cvrl/UNDBiometricsDatabase.html>
2. Bagdanov, A.D., Del Bimbo, A., Masi, I.: The Florence 2D/3D hybrid face dataset. In: Proceedings of Joint ACM Workshop on Human Gesture and Behavior Understanding, pp. 79–80 (2011)

3. Berretti, S., Del Bimbo, A., Pala, P.: 3D face recognition using iso-geodesic stripes. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(12), 2162–2177 (2010)
4. Berretti, S., Del Bimbo, A., Pala, P.: Sparse matching of salient facial curves for recognition of 3-D faces with missing parts. *IEEE Trans. Inf. Forensics Secur.* **8**(2), 374–389 (2013)
5. Bondi, E., Pala, P., Berretti, S., Del Bimbo, A.: Reconstructing high-resolution face models from Kinect depth sequences. *IEEE Trans. Inf. Forensics Secur.* **11**(12), 2843–2853 (2016)
6. Bookstein, F.L.: Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(6), 567–585 (1989)
7. Bowyer, K.W., Chang, K.I., Flynn, P.J.: A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition. *Comput. Vis. Image Underst.* **101**(1), 1–15 (2006)
8. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Robust expression-invariant face recognition from partially missing data. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3953, pp. 396–408. Springer, Heidelberg (2006). doi:[10.1007/11744078_31](https://doi.org/10.1007/11744078_31)
9. Cohen-Steiner, D., de Verdière, E.C., Yvinec, M.: Conforming Delaunay triangulations in 3D. *Comput. Geom.* **28**(2–3), 217–233 (2004)
10. Colombo, A., Cusano, C., Schettini, R.: Gappy PCA classification for occlusion tolerant 3D face detection. *J. Math. Imaging Vis.* **35**(3), 193–207 (2009)
11. Drira, H., Ben Amor, B., Srivastava, A., Daoudi, M., Slama, R.: 3D face recognition under expressions, occlusions, and pose variations. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(9), 2270–2283 (2013)
12. Faltemier, T.C., Bowyer, K.W., Flynn, P.J.: A region ensemble for 3D face recognition. *IEEE Trans. Inf. Forensics Secur.* **3**(1), 62–73 (2008)
13. Huang, D., Ardabilian, M., Wang, Y., Chen, L.: 3D face recognition using eLBP-based facial representation and local feature hybrid matching. *IEEE Trans. Inf. Forensics Secur.* **7**(5), 1551–1564 (2012)
14. Lu, X., Jain, A.K., Colbry, D.: Matching 2.5D face scans to 3D models. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(1), 31–43 (2006)
15. Passalis, G., Perakis, P., Theoharis, T., Kakadiaris, I.A.: Using facial symmetry to handle pose variations in real-world 3D face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(10), 1938–1951 (2011)
16. Porro-Muñoz, D., José Silva-Mata, F., Revilla-Eng, A., Talavera-Bustamante, I., Berretti, S.: 3D face recognition by functional data analysis. In: Bayro-Corrochano, E., Hancock, E. (eds.) *CIARP 2014*. LNCS, vol. 8827, pp. 818–826. Springer, Cham (2014). doi:[10.1007/978-3-319-12568-8_99](https://doi.org/10.1007/978-3-319-12568-8_99)
17. Savran, A., Alyüz, N., Dibeklioglu, H., Çeliktutan, O., Gökberk, B., Sankur, B., Akarun, L.: Bosphorus database for 3D face analysis. In: Schouten, B., Juul, N.C., Drygajlo, A., Tistarelli, M. (eds.) *BioID 2008*. LNCS, vol. 5372, pp. 47–56. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-89991-4_6](https://doi.org/10.1007/978-3-540-89991-4_6)
18. Wang, Y., Liu, J., Tang, X.: Robust 3D face recognition by local shape difference boosting. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(12), 1858–1870 (2010)