Expression-Insensitive 3D Face Recognition using Sparse Representation

Xiaoxing Li, Tao Jia
Electrical and Computer Engineering
Virginia Tech, USA
`{xxli,taojia}@vt.edu`

Hao Zhang
School of Computing Science
Simon Fraser University, Canada
`haoz@cs.sfu.ca`

Abstract

We present a face recognition method based on sparse representation for recognizing 3D face meshes under expressions using low-level geometric features. First, to enable the application of the sparse representation framework, we develop a uniform remeshing scheme to establish a consistent sampling pattern across 3D faces. To handle facial expressions, we design a feature pooling and ranking scheme to collect various types of low-level geometric features and rank them according to their sensitivities to facial expressions. By simply applying the sparse representation framework to the collected low-level features, our proposed method already achieves satisfactory recognition rates, which demonstrates the efficacy of the framework for 3D face recognition. To further improve results in the presence of severe facial expressions, we show that by choosing higher-ranked, i.e., expression-insensitive, features, the recognition rates approach those for neutral faces, without requiring an extensive set of reference faces for each individual to cover possible variations caused by expressions as proposed in previous work. We apply our face recognition method to the GavabDB and FRGC 2.0 databases and demonstrate encouraging results.

1. Introduction

Sparse representation, also known as compressed sensing [6, 7], has been applied recently to image-based face recognition [26] and demonstrated encouraging results. Under this framework, each face is represented by a set of features, which sufficiently characterize each individual. With the prior knowledge that faces of the same individual are similar to each other, a probe face can be considered as being well approximated by linearly combining the k reference faces of the same individual in the training set. In other words, a probe face is inherently k-*sparse* [6] on the basis of all n reference faces in the training set. Compared with conventional methods, recognition method based on the sparse representation framework has several unique ad-

vantages. First, by solving an L_0-norm minimization problem, the objective of the recognition method is to relate a probe face with the *minimum* number of reference faces. Considering that the task of face recognition is to find a single individual out of the entire dataset, this objective is naturally better suited for face recognition purposes. Second, due to the inherent *source-and-error separation* property, the presence of irrelevant features will not degrade recognition performance, as long as the remaining features are sufficiently informative. A strong empirical evidence is that the 2D face recognition method proposed in [26] can tolerate up to 40% occlusion and 70% data corruption. Third, in the sparse representation framework, increasing the number of individuals in the dataset in fact improves the sparsity and thus will not likely degrade recognition performance, which is in contrast to conventional methods [2, 24].

Despite the promising advantages, the use of sparse representation for face recognition does not eradicate obstacles posed by several practical issues, such as lighting, pose, and especially facial expressions, which tend to distort almost all the features and can thus compromise the sparsity of the representation. To achieve robustness to expressions, the authors in [26] postulated that a well-designed training set is needed such that the set of reference faces for each individual covers possible variations caused by expressions. Nevertheless, most existing databases do not fulfill this requirement. It may turn out to be either an unrealistic or expensive criterion to achieve in practice.

3D face data are less sensitive to lighting variation and pose change, which have been adopted either as an addition or a substitution to face images in recognition tasks [13]. However, 3D faces have their own difficulties. First, the absence of a consistent parameterization among face meshes or point clouds makes it impossible to directly obtain registered features with a uniform sampling pattern; this prevents appearance-based methods or the sparse representation framework from being applied. More importantly, facial expressions can cause severe global geometry variations, resulting in degraded recognition performance [3, 17] or heavy computational load [4, 12].

In this paper, we present a face recognition method based on sparse representation for recognizing 3D face meshes. Our main contributions come from two aspects. First, to enable the application of the sparse representation framework, we develop a simple yet effective uniform remeshing scheme to obtain a consistent sampling pattern across face mesh surfaces. Unlike previous works which rely on either high-level features or a planar uniform grid to characterize 3D face geometry, we impose uniform connectivity onto the 3D face surfaces for recognition which turns out to be a prerequisite to enabling the collection of registered low-level features and subsequent application of the sparse representation framework. Second, we develop a feature pooling and ranking scheme to improve the robustness of our method to severe expressions. This scheme eliminates the necessity of constructing an extensive set of reference faces for *each* individual, which we believe complements previous work in [26]. Our experimental results on GavabDB [18] and FRGC 2.0 [9] databases show that, using merely the most common low-level features, *e.g.*, mean curvatures at vertices, triangle areas and triangle inner angles, our approach achieves encouraging recognition rates.

2. Background and motivation

Most existing 3D face recognition methods are based on strategically extracted high-level features, *e.g.*, shapes of facial curves [23], concave and convex facial regions [3], partial face regions [14, 19], or deformation distance metrics [12]. In these methods, most efforts have been spent on extracting these features and the recognition performance generally depends on their reliability. In this work, however, we are interested in adopting low-level geometric features to face recognition for several reasons. First, collecting low-level features requires less sophisticated algorithms. Second, low-level features tend to be more reliable, since the collection process does not rely on pre-defined semantic extraction strategy. More importantly, the redundancy carried in the large number of low-level features can be naturally exploited by the sparse representation framework to achieve robustness against occlusion and data corruption.

However, collecting low-level features on 3D face surfaces is a non-trivial task, simply because samples of the original input can be placed at arbitrary locations, unlike the 2D case where a fixed Cartesian grid is employed. Due to the lack of a uniform sampling pattern, there is no natural correspondence between the low-level geometric features collected from different 3D faces. This motivates us to develop a remeshing scheme to embed a uniform connectivity onto all 3D face surfaces, and thus imposing a correspondence between any low-level features defined on matching triangles or vertices on the remeshings across 3D surfaces. In addition, as will be shown later, the collected low-level features are not sufficiently informative to
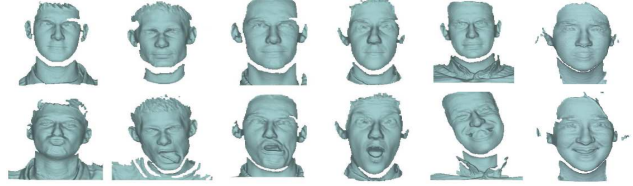

Figure 1. 3D face examples from GavabDB. Top row: neutral faces. Bottom row: expressioned faces. Faces in the same column belong to the same individual.

achieve satisfactory recognition performance under conventional appearance-based methods, *e.g.*, PCA [24]. It is the sparse representation framework that facilitates the use of low-level features for robust face recognition in our approach. Note that methods based on range images [17, 20] can also be viewed as imposing uniform sampling pattern onto 3D faces. However, the acquisition of range images involves an $\mathbb{R}^3 \to \mathbb{R}^2$ projection, which is not pose-invariant.

It is well-known that expressions can significantly distort facial geometry (Figure 1), and this has become the major limiting factor for robust 3D face recognition. To handle this issue, most existing works rely on the selection of certain high-level expression-invariant features [4, 12, 21, 25]. For example, the authors in [4] assume that face surfaces are isometric under expression and select bending-invariant canonical forms of face surfaces as the expression-invariant feature. Face recognition method based on sparse representation [26], despite its superior robustness to occlusion, cannot directly handle expressions, due mainly to the fact that distortions caused by expressions may occur over almost the entire facial area, whereas occlusions only affect partial regions. In this paper, we design a feature pooling and ranking scheme, which first collects a large set of low-level features and then, via a training process, ranks them according to their sensitivities to expressions. By choosing higher-ranked expression-insensitive features, our recognition method achieves robustness to severe expressions.

3. Method

We characterize each face by a *descriptor*, whose elements are a set of registered low-level geometric features. To enable the collection of such features, a uniform remeshing scheme is first applied. Then, we use a feature pooling and ranking scheme to obtain expression-insensitive descriptors by stacking higher-ranked features, which are finally used to recognize 3D faces under the framework of sparse representation.

3.1. Uniform remeshing

Consistent remeshing has been studied in previous works, *e.g.*, [22], for the purpose of blending shapes and animation. These methods are specially designed to achieve
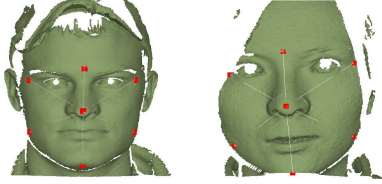
Figure 2. Facial mask. Left: an ideal mask. Right: an approximated mask, where the desired locations of the markers at the corner of the right eye and the chin tip are occluded, and thus the closest nearby vertices are selected.

Figure 3. Sample faces with data defects from GavabDB. From left to right: 1) - 2) occlusion by hair; 3) occlusion by head rotation; 4) occlusion by hand; 5) impulse noise around the chin caused by scanning artifacts due to shading; 6) under-sampled facial area.

robustness against shape or topology variations across input meshes, but at the expense of more costly computation. Our remeshing scheme, in contrast, is only targeted at face surfaces, which have a relatively stable shape. Computational load is, however, of more importance, especially when operating on large face databases. Bearing these in mind, we develop an efficient remeshing scheme for face meshes, which starts by building a face mask from 7 markers. Then, by an iterative subdivision and displacement procedure, remeshed faces at different resolutions can be obtained.

Mask: Our remeshing scheme starts by building a simple mask which contains 6 triangles and 7 vertices, as shown in Figure 2 (left). The 7 vertices are facial markers, which are defined as the nose tip, the center of the two eyebrows, the outer corners of the two eye sockets; the two points on the facial outline contour at the same level as the center of the mouth; and the chin tip. In fact, there exist algorithms capable of automatically detecting such markers, *e.g.*, [1]. However, the databases we are working on carry various data errors and deteriorations, as can be seen in Figure 3. Due to such defects, in this work we choose to manually select the 7 markers. As can be seen, the main purpose of the mask is to crop out and subsequently triangulate the desired facial region. Note that for real applications, some of the desired markers may be occluded. When this occurs, we place the corresponding markers at the closest available vertices. An example is shown in Figure 2 (right).

There are several rationales for choosing such a mask. First, the only non-boundary vertex in the mask has valence 6, i.e., the nose tip. In doing so, after subdivision, our remeshings will all possess regular connectivity in the interior. Such regularity offers several advantages in computing local geometry features, *e.g.*, curvatures [10]. Sec-
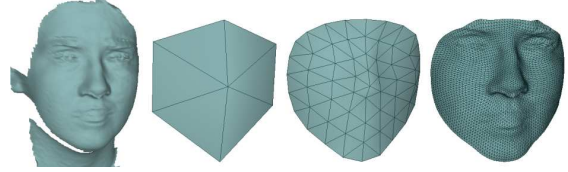

Figure 4. Remeshings. An original face mesh and its remeshings. From left to right: original mesh with 13709 vertices; level-0, 2 and 5 remeshings, with 7, 61, and 3169 vertices, respectively.

ond, our mask covers the desired facial area with a substantially small number of markers. Such a small number of well-defined markers minimize manual work and reduce the chance of marker misplacements.

Remeshing: Our remeshing scheme is inspired by the idea of displaced subdivision surfaces proposed by Lee et al. [15]. We modify, as well as simplify, the original algorithm in several aspects.

In Lee et al. [15], the so-called control mesh is a geometrically-close approximation to the original surface. Triangles can be co-planar subdivided for multiple levels and smoothed by an optimization process to obtain a domain mesh. In our case, however, we adopt the face mask as the control mesh. Such a simple control mesh is only a coarse approximation to the original face surface. More importantly, vertices in our control mesh are registered markers, which are not allowed to drift during the subsequent remeshing process. Therefore, we implement an iterative two-step process. In the first step, we perform Butterfly subdivision [27], which is inherently interpolatory, to generate the domain mesh. In the second step, vertices on the domain mesh are displaced onto the original mesh along their normal directions.

Note that after one iteration, the number of triangles is quadrupled and the displaced surface is a remeshing at a resolution one-level higher than that of the previous iteration. We refer to the face mask as level-0 remeshing. The displaced surface after the first iteration is called the level-1 remeshing, and so on. In Figure 4, we give an example of an original face mesh and its level-0, 2 and 5 remeshings.

Furthermore, to deal with data defects, such as holes, gaps and impulse noise, we introduce several additional criteria into the displacement step. First, when multiple intersections occur between an original mesh and the normal vector of a vertex on the domain mesh, we simply select the closest one. Second, even with a single intersection detected for a vertex, if the displacement exceeds a pre-defined threshold, we consider this as possibly missing data or impulse noise. We then use the linear average of the finite displacements for the one-ring neighbors to displace the vertex. Third, if a displacement of a vertex does not intersect the original mesh, implying the presence of a hole or gap,

the same procedure mentioned above, i.e., one-ring averaging over the neighborhood is applied. Finally, if all the displacements in a small neighborhood on the domain mesh do not intersect the original mesh or fail the threshold test, it is treated as an occluded area. The domain mesh overlaid with this area is then directly used in the remeshing.

3.2. Feature pooling and ranking

For face images, pixel intensity is the commonly used low-level feature. In 3D face surfaces, geometry information can be captured by various *types* of low-level features. Examples include the curvature at each vertex, the area of each triangle, the length of each edge, and so on. It is difficult to determine which type of features are more informative for face recognition. In addition, as described in our remeshing scheme, we obtain remeshed faces at multiple resolutions, which further increases the number of available features. Since features collected at different resolutions potentially carry complementary information, i.e., features collected from higher resolution remeshings capture more local geometric details, while those from lower resolution ones characterize more global geometry, it is again difficult to determine which resolution leads to more informative features.

All these available features form a pool. However, under the sparse representation framework, adopting all the features for recognition is unrealistic. More importantly, the majority of these features can be distorted to certain extent by facial expressions. Unlike occlusion, where source-and-error separation automatically excludes those corrupted features [26], distortions caused by expressions usually spread over all the features. It has been observed that such distortions can damage the sparsity of the representation and thus degrade recognition performance.

Although facial expressions are complicated, human faces are not fully elastic, implying that expressions only distort face geometry in finite categories [8]. An important observation from [16] is that there exist some low-level geometric features that are relatively insensitive to expressions. As shown in Figure 5, we observe that from a neutral face to a "laughing" face, the geodesic distance from the nose tip to the left eye changes only slightly. However, the geodesic distance from the nose tip to the chin tip is evidently increased. Another such sample is that the curvatures at the vertices on the cheeks of the "laughing" face change as facial muscles contract, yet the areas of most triangles on the cheeks do not change much due to the limited stretch of facial skin.

Based on the above observations, we can obtain a set of low-level features that are relatively insensitive to expressions through a training process. Specifically, for a training set containing m individuals and k reference faces for each individual, there are a total of $n = m \times k$ reference faces,
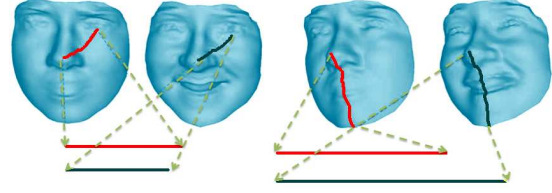

Figure 5. Impact of expression on geodesic distance between facial markers. Left: the distance changes slightly under expression. Right: the distance changes significantly under expression.

denoted by

$$\{f_{i,j} \mid i = 1, 2, \ldots, m, \; j = 1, 2, \ldots, k\}, \qquad (1)$$

where $f_{i,j}$ represents the jth face of the ith individual. After applying the aforementioned remeshing scheme, we collect a pool of features for each face, which may be of different types or collected at different resolutions. For example, the area of a particular triangle collected from a level-2 remeshing can be a feature. The curvature at a particular vertex from a level-3 remeshing is another feature. Denote the value of the tth feature on $f_{i,j}$ by $v_{i,j}^t$, the *confidence* of the tth feature, a measure of its sensitivity to expressions, is evaluated as

$$C^t = \frac{S_B^t}{S_W^t} = \frac{\sum_{i=1}^m (\bar{v}_i^t - \bar{v}^t)^2}{\sum_{i=1}^m \sum_{j=1}^k (v_{i,j}^t - \bar{v}_i^t)^2}, \qquad (2)$$

where

$$\bar{v}_i^t = \frac{1}{k} \sum_{j=1}^k v_{i,j}^t, \quad \bar{v}^t = \frac{1}{n} \sum_{i=1}^m \sum_{j=1}^k v_{i,j}^t. \qquad (3)$$

$\bar{v}^t$ is the mean value of the tth feature over all the references faces, while $\bar{v}_i^t$ is the mean value over the reference faces of the ith individual. We can see that the confidence defined in (2) has a form similar to Fisher's linear discriminant [2], where S_B^t and S_W^t can be viewed as the between-class and within-class scatters, respectively. The only difference is that in our work, the purpose of computing confidence is to evaluate a feature's sensitivity to expressions, whereas in [2], the purpose is to span a subspace by linearly combining all the features.

After ranking the features according to their confidence, we can stack higher-ranked features to form a descriptor, referred to as the *expression-insensitive descriptor* (EID). We believe EID's insensitivity to expressions helps improve the sparsity of the representation. In Figure 6, we give an experiment to demonstrate the effectiveness of using the EID. To recognize a "puffed" face, we collect mean curvatures at all the vertices on the level-3 remeshing to form the feature pool. In the first experiment, we use a subset of randomly selected features from the pool to form a descriptor.
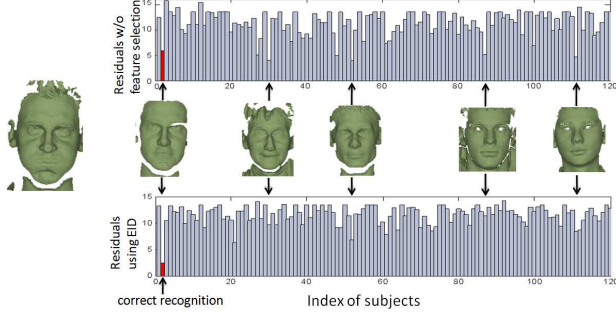
Figure 6. Effectiveness of feature ranking in recognizing an expressioned face (left). The bars represent the resulting residuals, (defined in (7)), between the probe face and each individual after applying the sparse representation framework. Upper row: using randomly selected features. Lower row: using the EID. As shown, the descriptor consisted of randomly-selected features results in insufficient sparsity and incorrect recognition, while the EID improves the sparsity and provides the correct recognition.

In the second experiment, an EID containing the same number of features is adopted. From Figure 6, we clearly see that the descriptor formed by randomly selecting features results in insufficient sparsity and leads to incorrect recognition, while the EID improves the sparsity and provide the correct result.

It is important to emphasize that when ranking features, we do not require reference faces of each individual to cover all the possible variations caused by expressions. Instead, we only require that these variations are covered by the *entire* training set across different individuals. This suggests that the confidence defined in (2) reflects solely the impact of expression on each feature, and is independent of the characteristics of individuals.

3.3. Recognition

Given a training set as (1), we compute a length-l EID for each reference face, denoted by

$$\mathbf{v}_{i,j} = [v_{i,j}^1, v_{i,j}^2, \ldots, v_{i,j}^l]^T, \tag{4}$$

where $(\cdot)^T$ denotes matrix transpose. Similarly, we compute the EID $\mathbf{v}_p$ for a given probe face f_p. The EIDs of reference faces in the training set can be organized into a matrix $\mathbf{A} \in \mathbb{R}^{l \times n}$, denoted by

$$\mathbf{A} = [\mathbf{v}_{1,1}, \ldots, \mathbf{v}_{1,k}, \mathbf{v}_{2,1}, \ldots, \mathbf{v}_{2,k}, \ldots, \mathbf{v}_{m,1}, \ldots, \mathbf{v}_{m,k}].$$

With both $\mathbf{v}_p$ and $\mathbf{A}$, the task of recognizing f_p under the framework of sparse representation is to solve the following L_0-norm minimization problem

$$\hat{\mathbf{x}}^{L_0} = \arg \min \|\mathbf{x}\|_0, \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{v}_p, \tag{5}$$

where

$$\mathbf{x} = [x_{1,1}, \ldots, x_{1,k}, x_{2,1}, \ldots, x_{2,k}, \ldots, x_{m,1}, \ldots, x_{m,k}]^T$$

is a length-n coefficient vector, with ideally at most k non-zero entries. That is, $\mathbf{v}_p$ is k-sparse over $\mathbf{A}$. However, (5) is well known as an NP-hard problem. Fortunately, in compressed sensing [6, 7], as long as the solution to (5) is known to be sufficiently sparse, an equivalent L_1-norm minimization

$$\hat{\mathbf{x}} = \arg \min \|\mathbf{x}\|_1, \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{v}_p, \tag{6}$$

can be solved as a good approximation to (5). Favorably, (6) can be solved by standard linear programming techniques. In fact, conventional recognition methods relying on Euclidean distances to find the nearest neighbors can be modeled as a minimization problem similar to (5), except that the objective to be minimized is the L_2-norm.

With the solution $\hat{\mathbf{x}}$ to (6), we can compute the residual between f_p and each individual as

$$r_i = \|\mathbf{v}_p - \sum_{j=1}^{k} \hat{x}_{i,j} \mathbf{v}_{i,j}\|_2. \tag{7}$$

The identity of f_p is then determined as the one with the smallest residual. Using L_2- instead of L_0-norm residuals has been proposed in [26]. To the best of our knowledge, this may be attributed to the fact that, due to the source-and-error separation property, L_0-norm minimization leads to a set of coefficients more robust to noise compared to L_2-norm minimization. With less impact from noise on the coefficients, it is beneficial to use L_2-norm residuals to emphasize any small difference among individuals in recognition, compared to the use of L_0-norm residuals.

4. Experiments and results

In this section, we present experimental results. We first describe the experimental setup, followed by a description of several types of low-level features used. Then, we test our recognition method and discuss the results.

4.1. Database and experimental setup

To the best of our knowledge, GavabDB [18] is the most expression-rich and noise-prone 3D face database currently available to the public. We test our method on this database, which contains 5 frontal face scans for each of the 60 captured individuals. To further evaluate the robustness of the sparse representation framework in a larger search space, we double the number of individuals from 60 to 120, by randomly selecting another 60 individuals from the well-known FRGC 2.0 database [9]. Since some individuals in FRGC 2.0 have less than 5 face scans, we confine our random selection to those individuals with at least 5 frontal scans. The combined dataset is denoted by

$$\{f_{i,1}, f_{i,2}, \ldots, f_{i,5} | i = 1, 2, \ldots, 120\}.$$

Figure 7. Two sets of faces in our dataset. Upper row: an individual from GavabDB. Lower row: an individual from FRGC 2.0. In both cases, the first two are neutral faces, while the other three carry expressions.

Specifically, the first 60 individuals are from GavabDB. For each individual, $f_{i,1}$ and $f_{i,2}$ are neutral faces, $f_{i,3}$ has a "random" expression, while $f_{i,4}$ and $f_{i,5}$ are faces with the expressions "laughing" and "smile", respectively. The next 60 individuals are from FRGC 2.0, where $f_{i,1}$ and $f_{i,2}$ are neutral faces, while $f_{i,3}$, $f_{i,4}$ and $f_{i,5}$ carry varied expressions. In Figure 7, we show the faces of two individuals from the two databases, respectively.

In our experiments, we use the leave-one-out scheme to construct the training and the testing sets. Specifically, each experiment consists of 5 sets of tests. For the first set of tests, the testing set is given by $\{f_{i,1} | i = 1, 2, \ldots, 120\}$ and the training set is given by $\{f_{i,2}, f_{i,3}, f_{i,4}, f_{i,5} | i = 1, 2, \ldots, 120\}$. The training sets and the testing sets in the other 4 sets of tests are constructed similarly. In doing so, each face in the dataset is used as the probe face for once. As a result, there are 600 probe faces being recognized in each experiment, for which the recognition rate using a random guess is as low as $1/120 = 0.83\%$.

4.2. Features

In our experiment, we collect four types of low-level features from the level-0 up to level-3 remeshings.

Angle: two registered inner angles of each triangle on a remeshing. In addition, all the dihedral angles between adjacent triangles are included. This type of feature is scale-invariant.

Triangle area: triangle areas of all the triangles on a remeshing. This type of feature is not invariant to scaling, and thus requires normalization.

Graph Distance between vertices: pair-wise graph distance between vertices. In particular, we compute graph distances from the vertex at the noise tip to all other vertices. This type of feature requires normalization.

Mean curvature: mean curvatures at all the vertices on a remeshing. It is computed as the average of the two principal curvatures [11]. This type of feature is again scale-invariant.

4.3. Recognition performance

Using these low-level features, we first compute the EID for each face. Then, using the EID, we apply the sparse representation framework as given in (6). We choose to use *L1-magic* [5] to solve (6). As for compressed sensing [6], to obtain the k-sparse coefficient, the *restricted isometric property* (RIP) needs to be satisfied. Assuming measurements are random Gaussian, one way to ensure the RIP is to impose the following inequality,

$$l \geq c\, k\, \log\left(\frac{n}{k}\right) \ll n, \qquad (8)$$

where c is a small constant. In our experiments, $n = 4 \times 120 = 480$, which is the number of reference faces in the training set, and $k = 4$, i.e., 4 reference faces for the same individual as a given probe face. Essentially, the recognition task is to find a 4-sparse representation of a probe face on the basis of 480 reference faces. In addition, we set $l = 160$, which is the number of features to be stacked into an EID. As can be seen, this setting satisfies (8).

4.3.1 Recognition using single feature type

To demonstrate the efficacy of our feature ranking scheme, we first recognize faces using a single feature type. For each type of feature, namely, angle, triangle area, graph distance, or mean curvature, we compare the recognition performances under two schemes. The first scheme uses a set of randomly selected (RS) features to form a descriptor. The second one uses the EID.

The recognition rates using different types of features under the two schemes are presented in Table 1 for neutral faces, expressioned faces and all faces, respectively. Based on these results, we have several important observations.

Table 1. Recognition rates using single feature type.

Feature Type	Descriptor	Neutral	Expression	Overall
Angle	RS	92.50%	86.67%	89.00%
	EID	93.33%	92.78%	93.00%
Triangle area	RS	90.00%	80.50%	84.17%
	EID	88.75%	86.39%	87.33%
Graph distance	RS	91.25%	81.94%	85.67%
	EID	92.50%	89.44%	90.67%
Mean curvature	RS	91.67%	85.83%	86.50%
	EID	92.83%	87.78%	89.50%

First, under the EID scheme, the overall recognition rate obtained using any of the four types of features is already satisfactory. The best overall recognition rate, using angle, is 93%. It is worth noting that the recognition rate for expressioned faces differs from that of neutral faces by at most 5.05% using mean curvature, and as small as 0.55% for angle. Considering the severe expressions in the dataset, this difference is strikingly small.
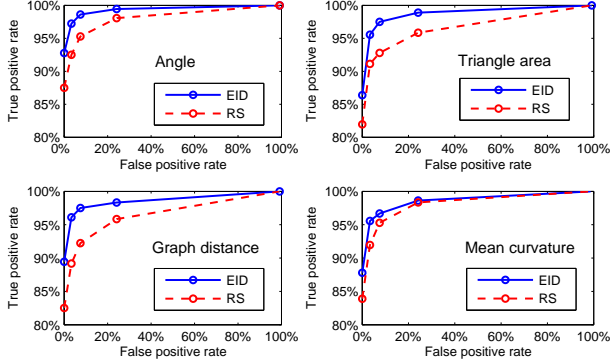
Figure 8. ROC of recognition rates for expressioned faces.

Figure 9. ROC of final recognition rates

Secondly, under both RS and EID schemes, the recognition rates using different types of features do not vary greatly, where angle leads to recognition rates slightly better than the others. For example, using the EID, the best overall recognition rate is 93%, obtained using angle, which is only 5.67% higher than the worse rate of 87.33%, obtained using triangle area. This observation agrees with the claim in [26] that as long as the sparsity of the representation is properly preserved, the exact choice of which type of feature to use is no longer critical.

For any feature type, the EID scheme outperforms the RS scheme in almost all cases, and the improvement is more significant when recognizing expressioned faces. Note that the recognition of neutral faces is also improved by using the EID, since it mitigates the adverse effect of including expressioned faces into the set of reference faces. An exception is the case for recognizing neutral faces using triangle area, which we believe is caused by certain random factors in the experiments. In Figure 8, we present the ROC curves for recognizing expressioned faces under the two schemes, using all four types of features. They clearly demonstrate the improvements by adopting EIDs. This result shows that our feature ranking scheme indeed improves the sparsity of the representation in the presence of expressions.

In addition, we should mention that the good recognition performance is mainly attributed to the adoption of the sparse representation framework. In fact, we recognized the faces using the four feature types based on PCA, and the best overall recognition rate, obtained using angle, is only 58.5%. This indicates that our low-level features are not sufficient to recognize faces using the conventional L_2-norm minimization method. Another point worth mentioning is that features collected from remeshing at multiple resolutions in fact facilitate the recognition. We conducted similar experiments using the EID scheme, where the feature pool is formed by only collecting features from the level-3 remeshing. Except for the case using triangle area, the recognition rate is lower than those reported in Table 1. Finally, our recognition rates demonstrate that severe
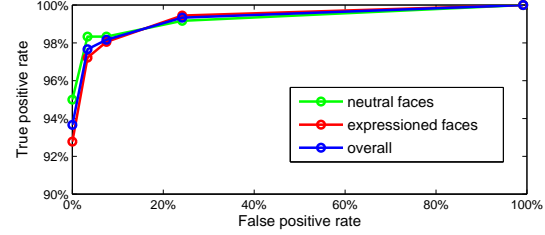
data defects, as shown in Figure 3, have little impact on the performance of our method. We believe that this can be attributed to the robustness of the sparse representation to occlusion and data corruption.

4.3.2 Recognition using all feature types

We conducted another experiment by collecting all the four types of features to form a combined feature pool. All the features in the pool are ranked together, where the top 160 features are used to form the EID. The ROC curves of the recognition rates using these EIDs, for neutral faces, expressioned faces and all faces, are shown in Figure 9. The marginal differences between these curves demonstrate the robustness of our proposed method to expressions.

Table 2. Comparison of recognition rates of one-best-matching using different methods.

methods	search space	accuracy
Our method	120 (reported rates are only for faces from GavabDB)	96.67% neutral 93.33% expressioned 94.68% overall
Moreno et al. [19]	60	90.16% neutral 77.9% expressioned
Mahoor et al. [17]	61	95% neutral 72.0% expressioned
Berretti et al. [3]	61	94% neutral 81% expression
Mousavi et al. [20]	61	91% overall

In Table 2, we compare the recognition rates of our method to several state-of-the-art results based on the GavabDB database. For a fair comparison, we only compare the rates of recognizing the GavabDB portion of our dataset, whereas our search space is 120 individuals, larger than others. Note that in [19] and [20], the dataset includes the two faces "looking up" and "looking down", which are not included in our test. Compared with existing methods, our proposed method has better performance in handling expressions. This improvement is due mainly to the efficacy of the sparse representation and is further enhanced by our feature ranking scheme. Note that similar high recognition rates have been obtained for the FRGC 2.0 database [14]. Since we only used part of this database, the performance of our method on the complete FRGC 2.0 is currently un-

available. Finally, we should point out that for the methods compared in Table 2, different data pre-processings are involved, whose purposes are similar to our marker selection procedure.

5. Conclusion and future work

We present a 3D face recognition method based on sparse representation. We demonstrate the necessity of establishing a uniform sampling pattern among 3D face meshes and develop a remeshing scheme to enable the collection of registered low-level feature and the subsequent application of the sparse representation framework. We further design a feature pooling and ranking scheme to achieve robustness to severe expressions. The satisfactory recognition rates using our proposed method demonstrate the efficacy of the sparse representation framework on 3D faces, as well as the effectiveness of our feature pooling and ranking scheme, which eliminates the need of an extensive set of reference faces for each individual in the training set. Possible future work includes incorporating high-level features into the descriptors and investigating their effectiveness.

References

[1] V. R. Ayyagari, F. Boughorbel, A. Koschan, and M. Abidi. A new method for automatic 3D face registration. In *IEEE CVPR*, pages 20–26, 2005.

[2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE PAMI.*, 19(7), 1997.

[3] S. Berretti, A. D. Bimbo, and P. Pala. 3D face recognition by modeling the arrangement of concave and convex regions. *Lecture Notes in Computer Science*, 4398(1):108–118, 2007.

[4] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Robust expression-invariant face recognition from partially missing data. In *ECCV*, 2006.

[5] E. Cand‘es, J. Romberg, and T. Tao. Caltech, L1-magic: http://www.acm.caltech.edu/l1magic/#links.

[6] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Information Theory*, 52(2), 2006.

[7] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52(4), 2006.

[8] P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto, 1978.

[9] P. J. Flynn. FRGC database v2.0, 2003. URL: http://bbs.bee-biometrics.org/.

[10] E. Grinspun, Y. Gingold, J. Reisman, and D. Zorin. Computing discrete shape operators on general meshes. *Computer Graphics Forum*, 25(3):547–556, 2006.

[11] B. Hamann. Curvature approximation for triangulated surfaces. *Geometric modelling*, pages 139–153, 1993.

[12] I. A. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE PAMI.*, 29(4):640–649, 2007.

[13] J. Kittler, A. Hilton, M. Hamouz, and J. Illingworth. 3D assisted face recognition: A survey of 3D imaging, modelling and recognition approachest. In *IEEE CVPR*, pages 114–114, 2005.

[14] I. C. Kyong, K. Bowyer, and P. Flynn. Multiple nose region matching for 3D face recognition under varying facial expression. *IEEE PAMI.*, 28(10):1695–1700, 2006.

[15] A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In *ACM SIGGRAPH*, pages 85–94, 2000.

[16] X. Li and H. Zhang. Adapting geometric attributes for expression-invariant 3D face recognition. In *Shape Modeling International*, pages 21–32, 2007.

[17] M. H. Mahoor and M. Abdel-Mottaleb. Face recognition based on 3D ridge images obtained from range data. *Pattern Recognition (in press)*, 42(3):445–451, 2009.

[18] A. B. Moreno and A. Sánchez. GavabDB: A 3D face database, 2004. URL: http://gavab.escet.urjc.es.

[19] A. B. Moreno, A. Sanchez, J. F. Velez, and F. J. Dkaz. Face recognition using 3D local geometrical features: PCA vs. SVM. In *Proc. of Int. Symp. on Image and Signal Processing and Analysis*, 2005.

[20] M. H. Mousavi, K. Faez, and A. Asghari. Three dimensional face recognition using svm classifier. In *Proc. of IEEE/ACIS Int. Conf. on Computer and Information Science*, pages 208–213, 2008.

[21] I. Mpiperis, S. Malassiotis, and M. G. Strintzis. Bilinear models for 3-D face and facial expression recognition. *IEEE Transactions on Information Forensics and Security*, 3(3):498–511, 2008.

[22] E. Praun, W. Sweldens, and P. Schroder. Consistent mesh parameterizations. In *ACM SIGGRAPH*, pages 179–184, 2001.

[23] C. Samir, A. Srivastava, and M. Daoudi. Three-dimensional face recognition using shapes of facial curves. *IEEE PAMI.*, 28(11), 2006.

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

[25] Y. Wang, G. Pan, and Z. Wu. 3D face recognition in the presence of expression: A guidance-based constraint deformation approach. In *IEEE CVPR*, pages 1 – 7, 2007.

[26] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE PAMI.*, 31(2):210–227, 2009.

[27] D. Zorin, P. Schröder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *ACM SIGGRAPH*, pages 189–192, 1996.