Adapting Geometric Attributes for Expression-Invariant 3D Face Recognition

Xiaoxing Li and Hao Zhang
GrUVi Lab, School of Computing Science
Simon Fraser University, BC, Canada
xli1,haoz@cs.sfu.ca

Abstract

We investigate the use of multiple intrinsic geometric attributes, including angles, geodesic distances, and curvatures, for 3D face recognition, where each face is represented by a triangle mesh, preprocessed to possess a uniform connectivity. As invariance to facial expressions holds the key to improving recognition performance, we propose to train for the component-wise weights to be applied to each individual attribute, as well as the weights used to combine the attributes, in order to adapt to expression variations. Using the eigenface approach based on the training results and a nearest neighbor classifier, we report recognition results on the expression-rich GavabDB face database and the well-known Notre Dame FRGC 3D database. We also perform a cross validation between the two databases.

1 Introduction

Face modeling and recognition has been one of the most intriguing and intensely researched topics in computer graphics and vision. Traditionally, 2D images are used for face recognition, due to ease of acquisition and the conceptual and computational advantages offered by the regular grid structure inherent in images. However, the 2D image of a person's face may vary drastically in appearance under different pose and illumination. Video streams that capture a face image sequence can provide more information to facilitate recognition, but they are not designed to remove the effects of pose and illumination. There is also the additional computational load for processing the image sequence data and the need for accurate registration between frames.

Recently, with the increasing ease with which 3D face models can be acquired and their relative insensitivity to changes in pose and illumination, much attention has been drawn to schemes which distinguish faces based on 3D geometry. So far, most geometry-based approaches rely on range images. A range image represents face geometry via depth values sampled over a regular 2D grid with respect to a reference plane, which is often hard to estimate due to the presence of noise and extraneous parts in a face capture [11]. Also, although most 2D face recognition methods can be directly applied to range images, which explains their popularity, certain intrinsic limitation with the image-based approach, e.g., the mis-alignment problem due to facial expression [24], is still preserved.

In this paper, we model face geometry using irregular triangle meshes. The free-form nature of meshes allows us to adjust sampling positions and density in a feature-sensitive way, which naturally supports more accurate alignment and more efficient use of triangles. Specifically, we remesh each face in the database into *uniform connectivity*. This remeshing effectively maps all the faces into the same feature space, via the explicit feature correspondence it provides. Thus a variety of geometric attributes can be collected and compared for 3D face discrimination.

The focus of our work is on accurate face recognition under moderate to severe facial expressions; refer to Figure 1 for a few examples of expressioned faces from the GavabDB 3D face database [27]. The adverse effect of facial expressions on face recognition is well known [10] and needs to be removed regardless of whether 2D or 3D face representation is being used. However, due to their nonlinear nature and the lack of an associated mathematical model, facial expressions are not easy to deal with.

The main contribution of this paper towards expression-invariant 3D face recognition is the idea that multiple geometric attributes derived from the face meshes can be utilized and even though they are all elementary, the key to improved performance lies in *adapting* these attributes to instabilities introduced by facial expressions and in effectively combining them; both can be realized through *training*. On one hand, we train for the element-wise weights for each individual attribute to obtain a set of *expression-insensitive signatures*. Each signature has its own ability to discriminate faces and can be further combined to form an optimized face recognition scheme. Our proposed 3D face recognition algorithm has been tested on the frontal

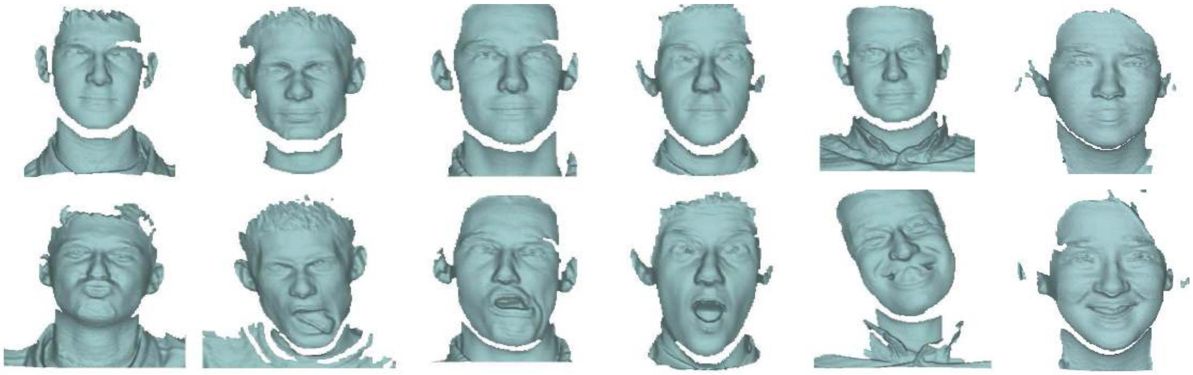


Figure 1. Examples of normal (top row) and expressioned faces from GavabDB [27].

face captures from the GavabDB database [27] which contains many severely expressioned faces. We are able to obtain recognition ratios of about 93% and 96%, depending on the number of reference faces used in recognition. On the Notre Dame (UNC) FRGC database [17], similar rates are reported and what is more important, when we apply the trained parameters from GavabDB to recognize faces in a *different* database, FRGC in this case, we can still achieve a corresponding recognition rates of about 85% and 95%. To the best of our knowledge, this type of cross-database validation results have not been reported before.

2 Related works

A typical face recognition system operates in three phases: face detection, feature extraction, and face recognition. Obviously, the quality of results from one phase can influence those from the following phases and an integrated approach would be most desirable. However, since each phase poses a difficult problem on its own which still lacks a completely satisfactory solution, a common and effective research practice is to focus on one particular phase for algorithm development and evaluation and assume that results from the previous phase(s) are in the best possible state.

Indeed, the majority of existing face recognition algorithms start with manual face alignment, trimming, or feature point selection. For example, in most methods based on range images, e.g., [32], the projection plane is manually selected and the facial region manually cropped out. When fitting a 3D morphable model to face images, markers that are needed in registration are often selected manually [6]. The facial surfaces are manually cropped out and rotated to fit in a Cartesian grid in the recent work of Bronstein et al. [10] for robust expression-invariant face recognition. There are even face databases which provide markers specified by human, e.g., [18]. On the other hand, there have also been tremendous amount of work which study face feature detection and registration alone, e.g., [16, 3]. In this paper, we focus on the recognition phase and rely on manually marked features to preprocess the face meshes.

Next we briefly survey previous works, focusing on purely geometric approaches applied to 3D face representations; more complete surveys on this topic can be found in [7, 33]. Most face recognition algorithms can be classified into *appearance-based* or *feature-based* approaches. The former measure the appearance of a face via quantities collected at densely-placed samples; some prominent examples include eigenfaces [36] and Fisherfaces [4], where recognition is performed in some low-dimensional subspace. One of the main factors which can compromise the performance of appearance-based methods is misalignment between the samples, e.g., under severe facial expressions [24]. Feature-based approaches rely on carefully extracted facial features, e.g., [12, 32], which are often more compact and support more accurate alignment. However, determining the right set of features is nontrivial and the thresholding problem [30] in many feature extraction methods can heavily influence the recognition results.

Any appearance-based 2D recognition technique can be easily adopted for range images, e.g., [2, 33]. Some of the geometric attributes considered include curvatures [12, 19] and extended Gaussian images [34]. Many feature-based methods have also been proposed for range images. A variety of facial features have been studied, e.g., multiple horizontal profiles [28], iso-contours on a face [32], and multiresolution range image gradients [29].

Face recognition using 3D meshes has drawn more attention lately. In earlier works, distances between face surfaces are computed via rigid iterative closest point schemes [26] or combined with Gaussian mixture models [14]. These methods cannot handle facial expressions however, which are non-rigid in nature. Of most relevance to our works are those by Bronstein et al. [8, 9, 10] on expression-invariant face descriptors. Next, we give a brief comparison between these works and the approach proposed in this paper.

The basic assumption made in the works of Bronstein et al. [8, 9, 10] is that facial expression is an *isometric* transformation, implying the preservation of area and geodesic distances over a face surface. Canonical forms are derived by embedding a face surface into a plane [8] or a 2-sphere [9]. The embeddings are treated as expression-invariant features and are subsequently used to register face textures for recognition. However, certain geometric information such as surface curvature has been lost in the embeddings. In addition, expressions around the mouth do not lead to isometry and would need to be treated separately.

The same authors subsequently propose the *partial embedding* method [10], where pairs of corresponding face patches are embedded into each other. As the region corresponding to an opened mouth is disregarded, there is no need for special handling. However, the computation required for comparing the patches would increase substantially. Even with hierarchical matching, intensive preprocessing is necessary for each new database. More importantly, there is no physiological support that facial expressions are isometric. In fact, our experiment show that surface areas and geodesic distances do vary under expressions but they are still important attributes for face discrimination.

Instead of relying on expression-invariant features, we utilize commonly used elementary geometric attributes, e.g., angles and geodesic distances, defined element-wise over the remeshed faces, akin to the appearance-based approach. These attributes are intrinsic to the surface and independent on the coordinate system chosen. As the remeshings are constructed in a feature-sensitive way, one may regard our approach as a *hybrid* one, taking advantage of both the appearance- and feature-based methods.

Note that we make no assumptions about invariance properties of our selected attributes under expression. Also, we do not assume expression invariance during the feature selection or remeshing process. Instead, by training the algorithm, the weighted descriptors will attempt to approach expression invariance by emphasizing the stable elements in the descriptors under expression and penalizing the unstable ones. The effectiveness of our approach relies on the quality of information offered by the training data.

3 The 3D face databases

We focus on the GavabDB 3D face database [27] in our studies as it offers special challenges in terms of data artifacts (see Figure 2) and severe facial expressions. The UND FRGC database, well-known in the face recognition community, is used to validate and cross-validate our training approach. GavabDB contains face meshes of 60 individuals, with 9 (numbered sequentially) per person captured under different settings. In particular, meshes 3 to 6 were captured under rotations resulting in partial profiles. Although the occluded parts may be reconstructed relying on face symmetry, this work is beyond the scope of our current study. We thus only select the remaining 5 faces per individual for our experiments: faces 1 and 2 are without expression; face 7 is "smiling"; the most challenging ones, faces 8 ("laughing") and 9 ("random gesture"), demonstrate pronounced facial expressions, where occlusions by the tongue or a hand can be present; see Figures 1 and 2. In the remainder of this paper, we would refer to mesh 1 as the *normal* face and the other $n = 4$ meshes as *expressioned* faces, where the second face would then be regarded as the one exhibiting a neutral expression.

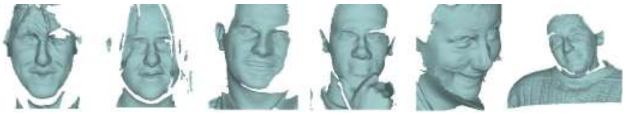


Figure 2. Some faces from GavabDB containing incomplete data, occlusion, noise, and extraneous parts (all dealt with in our work).

4 Uniform remeshing

Faces captured in GavabDB typically contain between 8K and 15K vertices and extraneous parts such as the shoulders and the neck. Prior to face recognition, each face model is remeshed, in a feature-sensitive manner based on manually selected face mask vertices, to have uniform connectivity and cover only the facial region. With face correspondence established through uniform mesh connectivity, any geometric feature computed per vertex, per feature point or per triangle can be used as an attribute or descriptor in modeling and comparing the appearances of the faces.

Note that the above remeshing problem has been studied before, e.g., by Praun et al. [31]. We have decided to develop our own simple procedure since we are working on a restricted class of shapes, i.e., faces, that are close to being planar; thus the patch construction step is simpler. On the other hand, our procedure implements a few simple heuristics to deal with incomplete data, by utilizing the symmetric nature of human faces, as well as impulse noise.

Feature point selection and the face mask: Our remeshed face covers the facial region from the eyebrow to the chin tip, excluding ears and the forehead, since they are typically occluded in the database, e.g., by hair. As argued in Section 2, we have decided to pick the 43 mask vertices manually, after which we construct a triangulated mesh containing 69 triangles, which we call the face *mask*, as shown in Figure 3(a). All the mask vertices are intended to mark facial features, e.g., they indicate the positions of the eyes,

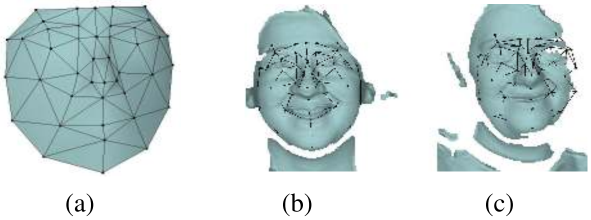


Figure 3. Face mask (a) and as it is fitted on face meshes, (b) and (c). In (c), the mask extends beyond the input mesh boundary.

nose, mouth, chin, etc., and define the face outline. We also place more mask vertices over feature-dense regions to capture the detailed features and define a connectivity so that most interior vertices have valance 5 or 6 to achieve regularity. With the mask, we can take advantage of the symmetry in human faces to extend the mask beyond the boundary of an input mesh containing missing data, as shown in Figure 3(c); this will help recover that missing data. Note that our mask is quite simple compared with those used in MPEG-4 [1] and for active appearance models [15], since for noisy face scans and a large database in our setting, manually or automatically registering a large set of feature points is unrealistic.

Domain mesh construction: Our remeshing procedure follows the idea of *displaced subdivision surfaces* [23], where the face mask serves as the control mesh. Subdividing the control mesh 3 times, we would obtain a domain mesh with 2297 vertices and 4480 triangles which is adequate for our face modeling and recognition tasks, as shown in the top row of Figure 4(f). However, in contrast to the algorithm in [23], where the control mesh is gradually optimized, our mask vertices (all are registered feature points) should not be allowed to drift. A natural choice would then seem to be Butterfly subdivision, which is interpolatory, but we have found that 3 levels of Butterfly subdivision may result in poor domain meshes, as shown in Figure 4(c); this is an artifact of the mesh connectivity and the low resolution of our face mask. Therefore, we have opted for the following heuristic for domain mesh construction: After applying Butterfly subdivision once to the control mesh, we run constrained non-shrinking Laplacian smoothing of Taubin [35] for 5 iterations with the mask vertices fixed. As subsequent Butterfly subdivision would introduce undesirable artifacts again, we simply run two levels of topological subdivisions (no change to geometry) to obtain the final domain mesh; these are illustrated in Figure 4.

Final mesh construction: We displace the vertices of the domain mesh along their normal directions to intersect the original mesh surface, where spatial division of the mesh surface is used to speed up the intersection search. When multiple intersections occur, we simply select the closest one. Moreover, we adopt three simple heuristics to deal with various artifacts in the input face data. First, if a displacement of vertex v does not intersect the original mesh, implying a case of missing data, then the (finite) displacements computed at the one-ring neighbors of v will be linearly interpolated to give the displacement at v. Furthermore, even with an intersection detected, if the displacement exceeds a threshold, we would still consider this to be a case of a missing data, or an impulse noise, and the same remedy will be applied. At last, if all the displacements in a small neighborhood on the domain mesh do not intersect the original mesh or fail the threshold test, then the domain mesh itself is used in the remeshing. Although these heuristics are rather primitive, for the moderate cases of missing data and impulse noise present in the databases we consider, they appear to be adequate as evidenced by the recognition performance we have been able to obtain. In Figure 5, we show some examples of our remeshing procedure at work.

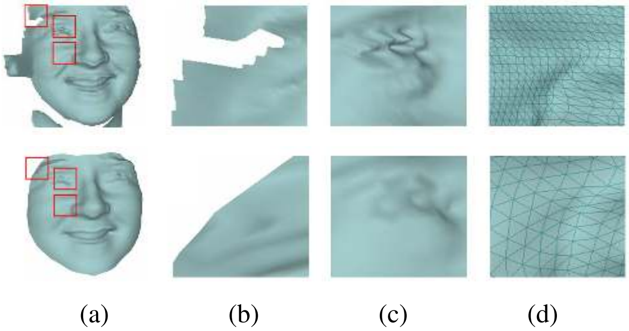


Figure 5. Bottom row: remeshing results; top row: original mesh. (a): Entire face. (b): Data completion. (c): De-noising. (d): Lower level of details and connectivity regularization.

5 Face recognition via training

In this section, we first list the set of elementary, intrinsic geometric descriptors adopted for our face recognition tasks. Then, we give a detailed description of the training procedure for both the element-wise weighting factor $\vec{\omega}$ and the attribute-wise weighting factor $\vec{\beta}$.

5.1 Intrinsic geometric descriptors

We select a set of elementary and commonly used *intrinsic* geometric descriptors. Mesh vertex coordinates, for example, are excluded since they are dependent on the coordinate system chosen. Compared to more sophisticated

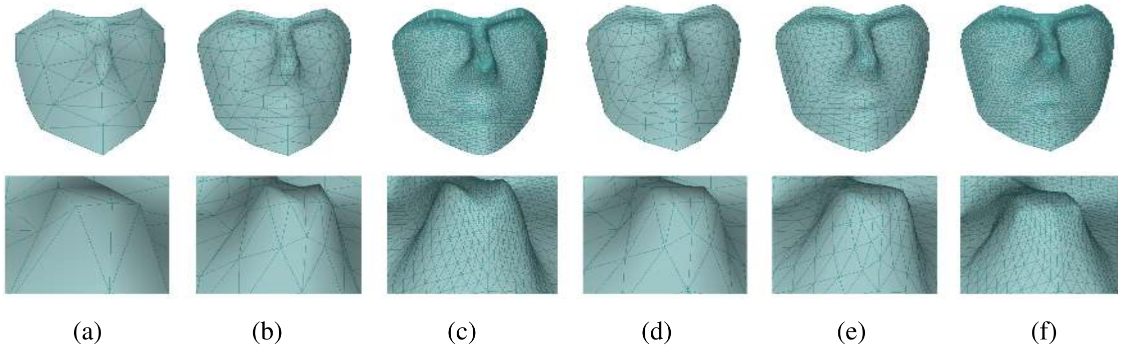


Figure 4. Domain mesh construction with bottom row a zoomed-in view of the nose tip. (a): Control mesh (mask). (b): After one level of Butterfly subdivision. (c): 3 levels of Butterfly subdivision lead to artifacts. (d): Constrained smoothing applied to (b). (e): Artifacts again after Butterfly subdivision applied to (d). (f): Domain mesh constructed after topological subdivision is applied to (d).

descriptors used in other works, e.g., [10, 32], our elementary descriptors allow for fast computation. Also, despite of their simplicity, they are shown to work effectively when equipped with an intelligent weighting scheme. In addition to descriptors derived from the remeshed faces, we also consider ones defined on the much lower-resolution face mask as they model the global face structures and are much more compact. We list our selected descriptors below.

Mask angles $\vec{A}$: The shape of the face mask represents a "skeleton" of the face it is fitted onto. We stack the inner angles of the triangles and dihedral angles of the mask mesh in some order to construct a column vector $\vec{A}$, which is scale-invariant and can be used to uniquely reconstruct the mask, given a scaling factor. For our face mask, $|\vec{A}| = 305$.

Pair-wise geodesic distances $\vec{L}$ on mask: Geodesic distances model the stretching of a surface. We stack the pair-wise geodesic distances between the 43 mask vertices to form a column vector $\vec{L}$, $|\vec{L}| = 903$, and normalize it to unit length. For ease of implementation, we use graph distances computed via Dijkstra's algorithm to approximate the true geodesic distances, which turns out to be quite adequate.

Triangle areas $\vec{T}$ on remeshed face: The column vector $\vec{T}$ stores the areas of all the 4,480 triangles in a remeshed face model, modeling stretching. This descriptor is again not scale-invariant and thus will be normalized into unit length.

Gaussian curvature $\vec{G}$ and mean curvature $\vec{M}$: We estimate the Gaussian and mean curvatures at the vertices following a parabolic fitting scheme [20], resulting in two column vectors of length 2,297, $\vec{G}$ and $\vec{M}$, respectively.

We can observe that none of the five descriptors above are invariant to facial expression. As an example, in Figure 6(b), we show a change of pair-wise geodesic distances caused by an opened mouth, when the face of the same individual changes from "normal" to "laughing". It is worth noting however that many descriptor elements do remain stable under expression, e.g., the pair shown in Figure 6(a).

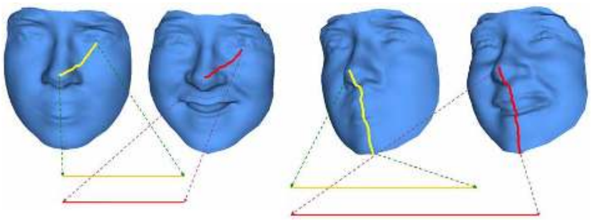


Figure 6. Effect of facial expression on our descriptor. Stable (left) and unstable (right) geodesic distances under "laughing".

5.2 Expression-insensitive signatures

We stipulate that the stable elements in the descriptors should be counted upon for discriminating faces between different individuals while variations of the unstable elements under facial expression would render them unreliable for the task. Therefore, we use a set of element-wise weights, obtained via training, to place more emphasis on the stable, or as we refer to as expression-insensitive, elements, and to suppress the effect of expression-sensitive elements. For brevity, the resulting descriptors will simply be referred to as expression-insensitive *signatures*.

In the following, we use generic notations $\vec{X}$ and $\vec{X}^*$ to represent any of the five un-weighted and weighted descriptors, $\vec{A}$, $\vec{L}$, $\vec{T}$, $\vec{G}$, $\vec{M}$ and $\vec{A}^*$, $\vec{L}^*$, $\vec{T}^*$, $\vec{G}^*$, $\vec{M}^*$, respectively, and the weight vector is denoted by $\vec{\omega}$. Superscripts index into the elements of a descriptor, e.g., $\vec{X}^{*(k)} = \vec{X}^{(k)} \cdot \vec{\omega}^{(k)}$, and subscripts label individuals and faces, e.g., $\vec{X}_{i,j}$ is a descriptor for the j-th face of individual i, where $1 \leq i \leq l$, l is the number of individuals in the training set, and $0 \leq j \leq n$. Note that $j = 0$ corresponds to the normal face.

Our training procedure works on a set of training faces for which the identifies are known. Face recognition is carried out using the commonly used nearest neighbor classifier based on Euclidean distances between face descriptors in eigenspace. One way to obtain the best weight vector $\vec{\omega}$ is via explicit optimization, minimizing the average Euclidean distance between the weighted descriptors of faces belonging to the same individual, i.e., intra-group distances, while maximizing the average inter-group distances. This is a large-scale (due to some descriptor sizes), non-convex, and constrained (e.g., requiring partition of unity of the weights) quadratic problem that is hard to solve. Instead, inspired by [22], we simply fit a function to $\vec{\omega}$ and optimize for the small set of parameters defining the function.

Weighting function: We propose a simple one-parameter weighting function $\vec{\omega}(\alpha)$, motivated by the notion of Mahalanobis distance [25] in its diagonal form. Specifically, the stability of a descriptor element is measured by the variance of its change under expression. We call the difference between the descriptors of the j-th expressioned face and the normal face, $\vec{R}_{i,j} = \vec{X}_{i,j} - \vec{X}_{i,0}$, an *expression residual*. Given a value of the free parameter α, the weight applied to the k-th element of the descriptor $\vec{X}$ when computing a distance from a probe face to a reference face is given by

$$\vec{\omega}^{(k)}(\alpha) = \Big[\sum_{i=1}^{l} \sum_{j=1}^{n} (R_{i,j}^{(k)} - \bar{R}^{(k)})^2 \Big]^{-\alpha}, \qquad (1)$$

where, $\bar{R}^{(k)}$ is the mean of the k-th element of all the expression residuals of the ln expressioned faces. The parameter α determines the degree of penalty for the unstable elements and it will be determined through training.

Training: In the *training stage*, we use faces in the training set as both training and testing data to simulate the recognition task and determine the α value. In the later *testing stage*, to ensure cross validation, faces in the training set and those used for testing are always *non-overlapping*.

In the training stage, we select the normal faces in the training set as reference faces, and use the remaining expressioned faces as probe data. We collect descriptors $\vec{X}$ for both reference faces and probe faces. For each value of α considered, we obtain a weighting vector $\vec{\omega}$ by (1). Then we project the weighted signatures into the eigenspace spanned by all the $\vec{X}^*$'s, and recognize the probe faces using a nearest neighbor classifier. The training result for α is the one which gives the best recognition rate among all probings. In the testing stage, given a testing face, we can incorporate the trained weights to acquire a collection of expression-insensitive signatures, $\vec{A}^*$, $\vec{L}^*$, $\vec{T}^*$, $\vec{M}^*$, and $\vec{G}^*$, where $\vec{A}^{*(k)} = \vec{A}^{(k)} \cdot \vec{\omega}^{(k),A}$, $1 \leq k \leq |\vec{A}|$, etc.

Finally, let us point out that although the training procedure is meant to remove the adverse effect of facial expression for face recognition, we do not and will unlikely be able to isolate the factor of facial expression from all other factors that may degrade recognition performance. These latter factors may include imperfect feature point selection (even done manually, errors can still be introduced) or noise and missing data that are not adequately repaired in our remeshing procedure. Thus as a side product, the trained weights will take care of the adverse effects generated by these unstable factors as well. We only need to ensure that the selected set of training faces is sufficiently representative so that the trained parameters will not be biased and result in poor recognition results during cross validation.

5.3 Combining signatures

It is quite conceivable that the five signatures $\vec{A}^*$, $\vec{L}^*$, $\vec{T}^*$, $\vec{M}^*$, and $\vec{G}^*$ are correlated. But at the same time, each signature may possess certain unique aspect in accomplishing face discrimination. Therefore, it would be beneficial to consider a combination of multiple signatures so as to improve recognition performance further. If we had known the correlations between the signatures, we would use them to drive a weighting scheme. However, since the lengths of our signatures all differ, it is difficult to compute the correlations analytically. That being said, note that our goal is only to obtain a set of attribute-wise weights to achieve high-quality recognition, thus we can combine the *distance matrices* derived from the signatures instead and rely on training to determine an approapriate weight vector $\vec{\beta}$, a 5-tuple forming a partition of unity, to apply to the distance matrices.

Specifically, for each face in the training set, we obtain a collection of signatures. Take a particular signature $\vec{X}$ for all the l reference faces and ln probe faces in the training data, we first compute the eigenvectors of the covariance matrix of the signatures and use them to span an eigenspace $\mathcal{S}^X$. We project these signatures into $\mathcal{S}^X$ where we can derive a matrix $\mathcal{D}^X \in \mathbf{R}^{l \times ln}$ containing pairwise distances between the reference and probe faces. If two signatures discriminate faces similarly, their distance matrices will be similar as well. Thus the correlation between the distance matrices, all having the same dimensionality $l \times ln$, can be used to approximate the correlations between their corresponding signatures. Consequently, we do not combine the signatures but combine their corresponding distance matrices instead. The weight vector $\vec{\beta}$ can be trained in the same way as for $\vec{\omega}$. We again recognize faces using a nearest neighbor classifier, which is now based on the $\vec{\beta}$-weighted combinations of the distance matrices.

6 Experiments and results

In this section, we report experimental results from our training and face recognition algorithms applied to two databases, the GavabDB [27] and the UND FRGC database [17]. For the faces considered in each database, we partition them into two groups G_1 and G_2 of equal size. Two sets of recognition tests are conducted. First, we use G_1 for training and G_2 for testing and then we reserve their roles. In each set of tests, we apply two test schemes. The first is called *Normal-Reference* (NR) scheme, where we select all the normal faces as reference faces and the remaining expressioned faces as probes. We recognize the probe faces by matching each of them to the closest reference face. The second scheme is the well-known *Leave-One-Out* (LOO) scheme [21], where we select one face for each individual as the probe and use the remaining faces to form a set of reference faces. A probe face is recognized as belonging to an individual if it has the smallest average distance to the reference faces of that individual.

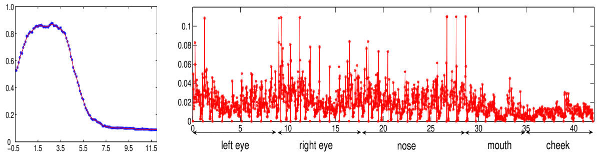
6.1 Results on the GavabDB database

We consider the 300 frontal face captures in GavabDB, 5 faces per individual, of all the 60 individuals contained in the database. Group G_1 contains 150 faces of individuals No. 1 to 30 and G_2 contains the remaining 150 faces. Thus for a test in the NR scheme, there will be 120 face recognitions performed. While for the LOO scheme, 150 recognition tests will be conducted.

6.1.1 Recognition with independent signatures

Training results on $\vec{\omega}$: In our experiment, we search for α in the range of $[-0.5, 10]$ with a step size of 0.1. Using eigenspaces of dimension 100, the time taken to train for a particular α in our setting is several minutes. In Fig. 7(a), the curve plots the recognition rates against α in the training stage, for the attribute $\vec{L}$, where group G_2 is taken as training set. Note that the curve is smoothed using a Gaussian to remove minor fluctuations. Due to space limitations, results for the other four attributes are not shown but they exhibit similar trends. In Fig. 7(b), we plot the trained weights for $\vec{L}$. For better visualization, we partition the set of feature points (mask vertices) into five groups and arrange them in order along the horizontal axis. As can be seen, the plots roughly indicate that lengths of geodesic paths going into or out of the mouth and cheek regions are relatively unstable and assigned with smaller weights.

The best α values obtained from the training stage are given in Table 1. As we can see, they are quite consistent from the two rounds of training, except for α^L. For this attribute, we can see from the curve shown in Figure 7(a) that there is a plateau around where the maximum recognition rate occurs, encompassing the two obtained α^L values 1.9 and 2.6. This indicates that varying the value of α in that small range will not lead to significant performance degradation. Thus overall, our trained α values are sufficiently general to be adopted for the face recognition tests.



(a) (b)

Figure 7. G_2-trained results for weights $\vec{\omega}^L$. (a): Recognition rate in the training stage, plotted against α. (b): Plot of trained weights; vertex grouping is shown along the horizontal axis.

Table 1. Training results for GavabDB.

		$\vec{A}$	$\vec{L}$	$\vec{T}$	$\vec{M}$	$\vec{G}$
G_1	α	1.2	1.9	1.0	1.3	1.2
G_2	α	1.3	2.6	1.1	1.6	1.4
G_1	$\vec{\beta}_1$	0.45	0.55	0	0	0
	$\vec{\beta}_2$	0.56	0.20	0	0.24	0
G_2	$\vec{\beta}_1$	0.50	0.50	0	0	0
	$\vec{\beta}_2$	0.35	0.40	0	0.25	0

Testing results: In Table 2, we report the recognition rates obtained in the testing stage. Clearly, the weighted descriptors consistently outperform their un-weighted counterparts. The two shortest descriptors, $\vec{A}^*$ and $\vec{L}^*$, returned the best results. Although the much longer area descriptor gave respectable rate, it contains a great deal of redundancy. The two curvature-based descriptors performed rather poorly, as they are quite sensitive to small surface variations. It is also worth noting that the recognition rates in Table 2 are close to the best rates for $\vec{A}$, $\vec{L}$, and $\vec{T}$ in the training stage, but not the other two, which further indicates that the curvature-based descriptors are unsuitable, at least independently, for expression-invariant face recognition.

6.1.2 Recognition with combined signatures

Training results on $\vec{\beta}$: We search for $\vec{\beta}$ in the unit 5-dimensional cube with a step size of 0.1 per dimension. The primitive exhaustive search only takes several minutes to complete and from these training results, we identify the best combination of two, three, four, and five descriptors. Regardless of whether G_1 or G_2 is used as the training set,

Table 2. Testing recognition rates, as $\%$ of correct matchings, using independent descriptors. "W-": trained $\vec{w}$ applied.

		$\vec{A}$	$\vec{L}$	$\vec{T}$	$\vec{M}$	$\vec{G}$
G_1	NR	68.33	60.00	33.33	22.50	8.33
	W-NR	89.17	81.67	63.33	65.00	50.00
	LOO	83.33	68.00	44.67	26.67	8.00
	W-LOO	93.33	88.67	77.33	64.67	41.33
G_2	NR	63.33	50.83	36.67	15.83	7.50
	W-NR	81.67	77.50	62.50	63.33	41.67
	LOO	83.33	62.00	47.33	18.67	6.67
	W-LOO	91.33	90.00	83.33	73.33	33.33

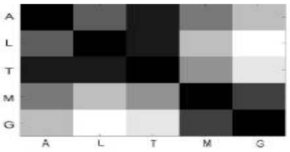


Figure 8. Mahalanobis distance between distance matrices of different signatures.

our results indicate that the best combination of two signatures are $\vec{A}$ and $\vec{L}$, which happen to be the shortest descriptors as well. The best combination of three come from $\vec{A}$, $\vec{L}$, and $\vec{M}$. Combinations of more attributes do not provide further performance gains. The trained weights $\vec{\beta}_1$ and $\vec{\beta}_2$ for the best 2- and 3-combinations are given in Table 1.

Instead of exhaustively searching for a best combination, we can seek guidance from the best recognition rates of the individual signatures in the training stage and the correlation between the distance matrices $\mathcal{D}^A$, etc., defined in Section 5.3. In Figure 8, we show a gray-scale plot of the Mahalanobis distances between the five distance matrices, where darker coloring indicates stronger correlation. As we can see, the best performing signatures $\vec{A}$ and $\vec{L}$ are relatively uncorrelated, compared with $\vec{A}$ and $\vec{T}$, for example. Thus it is reasonable to consider $\vec{A}$ and $\vec{L}$ as the best 2-combination. On the other hand, $\vec{M}$ and $\vec{G}$ are highly uncorrelated with the other three signatures but they perform poorly as independent signatures, thus it would be best to combine one of them with $\vec{A}$ and $\vec{L}$ to form a good 3-combination.

Final recognition results: Having obtained both weighting vectors $\vec{\omega}$ and $\vec{\beta}$, we can conduct the final recognition test in our testing stage. Each descriptor $\vec{X}$ of a test face is weighted by $\vec{\omega}^X$ to obtain the corresponding signature, which is projected into the proper eigenspace. The resulting distances, one per signature, between the test face and the reference faces, are combined using the trained $\vec{\beta}$. A nearest neighbor classifier is used to recognize the test face. The final recognition results are given in Table 3, where for comparison purposes, we also provide recognition rates using eigenfaces applied to the rigid surfaces, represented by the vertices of our remeshed face models. For rigid alignment, we apply ICP [5] which is initialized with the nose tips and major axes of the faces aligned first.

Table 3. Final recognition rates for GavabDB.

		Rigid surface	$\vec{A}^{*}$+$\vec{L}^{*}$	$\vec{A}^{*}$+$\vec{L}^{*}$+$\vec{M}^{*}$
NR	G_1	52.50%	92.50%	93.33%
	G_2	50.83%	91.67%	95.00%
	Avg.	51.67%	92.09%	**94.17%**
LOO	G_1	65.33%	96.00%	93.33%
	G_2	65.33%	98.00%	97.33%
	Avg.	65.33%	**97.00%**	95.33%

As we can see, the facial expressions in the test faces have significantly degraded the performance of the rigid eigenface approach while our approach results in far superior recognition rates. It is also interesting to observe that in the NR scheme, the combination of $\vec{A} + \vec{L} + \vec{M}$ performs better than $\vec{A}+\vec{L}$. However, in the LOO scheme, the inverse is true. It would appear that when expressioned faces, which tend to possess noisier curvature distributions, are included as reference faces, as in LOO, utilizing curvature attributes would have an adverse effect on face recognition.

Finally, we point out results from some other works on the GavabDB database. In [27], the authors of GavabDB develop a feature-based recognition method and the recognition rate (for single best matching) reported is 78%. More recently in [13], the face scans are projected into 2D range images to which a PCA approach is applied. The best recognition rate achieved using the NR and LOO schemes are 86.23% and 92.04%, respectively. In this work, as in our experiments, only the frontal face captures were tested.

6.2 Results on the UND FRGC database

The FRGC 3D face database (v2.0) [17] was captured by the University of Notre Dame and distributed in the Face Recognition Grand Challenge (FRGC). The faces therein are of higher resolution than those from GavabDB, but they are typically more complete and contain less expression. The first face for each individual is a normal face and the other 5 exhibit some level of expressions. Some sample faces and our remeshing results are given in Figure 9. We select 180 faces for the first 30 individuals, 6 per individual, from FRGC; they are divided into two groups as before.

In Table 4, we report the obtained α and $\vec{\beta}$ on FRGC from the training stage. Comparing these results with those in Table 1 for GavabDB, we see that they are relatively con-

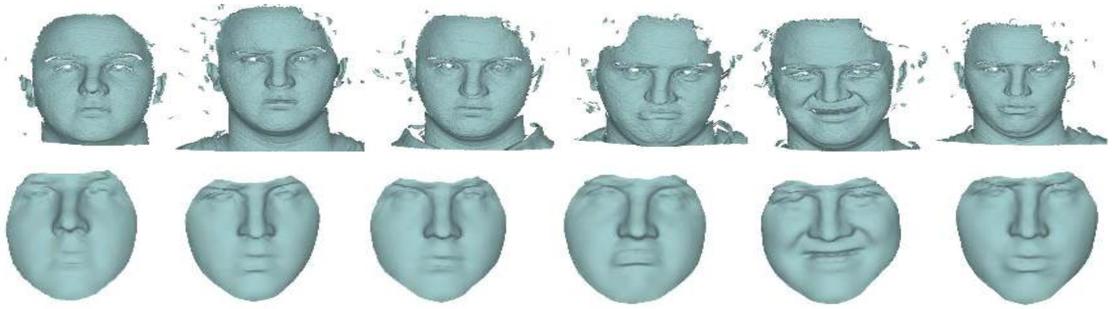


Figure 9. Sample faces from the UND FRGC database [17] and remeshing results.

sistent. When we conduct testing on the FRGC database, the average recognition rate for the NR and LOO schemes are 96.67% and 98.89%, respectively, where the combination of $\vec{A}+\vec{L}$ is used. The corresponding recognition rates for the best 3-combination are 94.65% and 99.45%, respectively. These higher rates may be explained by the more friendly test data in FRGC as well as a smaller test size.

Table 4. Training results for FRGC.

		$\vec{A}$	$\vec{L}$	$\vec{T}$	$\vec{M}$	$\vec{G}$
G_1	α	1.2	3.6	0.8	1.4	1.2
G_2	α	1.3	3.2	1.0	1.2	1.1
G1	$\vec{\beta}_1$	0.62	0.38	0	0	0
	$\vec{\beta}_2$	0.41	0.30	0	0.29	0
G2	$\vec{\beta}_1$	0.60	0.40	0	0	0
	$\vec{\beta}_2$	0.41	0.33	0	0.26	0

To examine the issue of over-fitting and further demonstrate the robustness of our algorithm, we conduct a cross validation *between* different databases. The corresponding recognition rates on the 180 faces from FRGC, when the two groups of 150 training faces from the GavabDB database are used as training data, are given in Table 5.

Table 5. Final cross-DB recognition rates.

	NR		LOO	
	$\vec{A}^{*}$+$\vec{L}^{*}$	$\vec{A}^{*}$+$\vec{L}^{*}$+$\vec{M}^{*}$	$\vec{A}^{*}$+$\vec{L}^{*}$	$\vec{A}^{*}$+$\vec{L}^{*}$+$\vec{M}^{*}$
G1	82.67%	78.67%	94.44%	87.78%
G2	88.00%	87.33%	96.67%	93.89%
Avg	**85.34%**	83.00%	**95.56%**	90.84%

It is worth noting here that in both the NR and LOO schemes, the combination of $\vec{A}^{*}+\vec{L}^{*}$ consistently outperforms the best 3-combination. This can again be attributed to the presence of noise in the signature $\vec{M}^{*}$, whose variation is even harder to model when faces from a different database are used as training data.

7 Conclusion and future works

Realizing the relative insensitivity of captured 3D geometry to pose and illumination, we propose an approach for 3D face recognition to achieve robustness against facial expression. Instead of deliberately constructing expression-invariant face features, we rely on elementary geometric descriptors. The key is to adapt them to instabilities introduced by facial expression and to effectively combine them to achieve optimized performance. Both can be accomplished via training which provides intelligent choice of weights to filter and combine the descriptors. The framework proposed is flexible in that it allows for the incorporation of additional descriptors. It would be especially interesting to identify compact and complementary descriptors for improved expression-invariant face recognition.

An unsatisfying aspect of our current work is manual feature selection, which can be tedious and error-prone. In the near future, we will replace this step by an automatic feature extraction algorithm and examine the robustness of our face recognition scheme against possible errors introduced. To further validate the effectiveness of our algorithm, the proposed system should be tested on other databases that contain a larger number of individuals.

Finally, our work confirms yet again the power of utilizing prior knowledge in shape analysis, a simple and effective idea. We would like to apply this idea in other settings where parameter tuning is necessary but the choices made so far have been ad-hoc. One such example is the weighting of factors that determine the salience of a segmented part [37]. We are also interested in looking beyond face recognition and examining how training can be helpful in tackling the more general shape retrieval problem.

Acknowledgement: The authors would like to acknowledge the use of the UND FRGC 3D Face Database (v2.0) and the generous help provided by Dr. Patrick J. Flynn from UND. The GavabDB face database used in the paper is due to Moreno and Sánchez [27]. This research is supported by an NSERC Grant (no. 611370) from the second author.

References

[1] MPEG Home Page: http://www.chiariglione.org/mpeg/.

[2] B. Achermann, X. Jiang, and H. Bunke. Face recognition using range images. In *Proc. Int. Conf. on Virtual Systems and MultiMedia*, pages 129–136, 1997.

[3] V. R. Ayyagari, F. Boughorbel, A. Koschan, and M. Abidi. A new method for automatic 3d face registration. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 20–26, 2005.

[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7), 1997.

[5] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2), 1992.

[6] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(9), 2003.

[7] K. Bowyer, K. Chang, and P. Flynn. A survey of approaches to three-dimensional face recognition. In *Proc. Int. Conf. on Pattern Recognition*, pages 358–361, 2004.

[8] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Expression-invariant 3d face recognition. In *Proc. Audio- and Video-based Biometric Person Authentication (AVBPA), Lecture Notes in Comp.*, pages 62–69, 2003.

[9] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *Int. Journal of Computer Vision (IJCV)*, 64(1), 2005.

[10] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Robust expression-invariant face recognition from partially missing data. In *Proc. European Conf. on Computer Vision*, 2006.

[11] G. E. C. Hesher, A. Srivastava. A novel technique for face recognition using range imaging. In *Proc. of Signal Processing and Its Applications*, pages 201–204, 2003.

[12] J. Cartoux, J. T. Lapreste, and M. Richetin. Face authentification or recognition by profile extraction from range images. In *Workshop on Interpretation of 3D Scenes*, pages 194–199, 1989.

[13] M. Celenk and I. Aljarrah. Interal shape-deformation invariant 3d surface matching using 2d principal component analysis. In *Proc. of SPIE-IS&T Electronic Imaging, SPIE*, volume 6056, pages 118–129, 2006.

[14] J. Cook, V. Chandran, S. Sridharan, and C. Fookes. Face recognition from 3d data using iterative closest point algorithm and gaussian mixture models. In *Proc. Int. Symp. on 3D Data Processing, Visualization and Transmission*, pages 502–509, 2004.

[15] T. Cootes, G. J. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6), 2001.

[16] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models — their training and application. *Comput. Vis. and Image Understand*, 61, 1995.

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

[18] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. Yale face database B. URL: http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html.

[19] G. Gordon. Face recognition based on depth and curvature features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 108–110, 1992.

[20] B. Hamann. Curvature approximation for triangulated surfaces. In *Computing. Supplementum*, pages 139–153, 1993.

[21] M. Kearns and D. Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. In *Neural Computation*, volume 11, pages 1427–1453, 1999.

[22] N. Kruger. An algorithm for the learning of weights in discrimination functions using a priori constraints. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7), 1997.

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

[24] X. Li, G. Mori, and H. Zhang. Expression-invariant face recognition with expression classification. In *Proc. Canadian Conf. on Computer and Robot Vision*, pages 77–83, 2006.

[25] K. V. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. 2000.

[26] G. Medioni and R. Waupotitsch. Face recognition and modeling in 3D. In *IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures*, pages 232–233, 2003.

[27] A. B. Moreno and A. Sánchez. Gavabdb: A 3d face database, 2004. URL: http://gavab.escet.urjc.es.

[28] G. Pan and Z. Wu. 3D face recognition from range data. *Int. Journal of Image and Graphics*, 5(3), 2005.

[29] S. Pansang, B. Attachoo, C. Kimpan, and M. Sato. Invariant range image multi-pose face recognition using gradient face, membership matching score and 3-layer matching search. *IEICE Trans. on Information and Systems*, 88(2), 2005.

[30] B. Park, K. Lee, and S. Lee. Face recognition using face-arg matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(12), 2005.

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

[32] C. Samir, A. Srivastava, and M. Daoudi. Three-dimensional face recognition using shapes of facial curves. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(11), 2006.

[33] A. Scheenstra, A. Ruifrok, and R. C. Veltkamp. A survey of 3d face recognition methods. In *Lecture Notes in Computer Science*, pages 891–899, 2005.

[34] H. Tanaka, M. Ikeda, and H. Chiaki. Curvature-based face surface recognition using spherical correlation. In *Proc. IEEE Conf. Automatic Face and Gesture Recognition*, pages 372–377, 1998.

[35] G. Taubin. Curve and surface smoothing without shrinkage. In *Proc. Int. Conf. on Computer Vision*, pages 852–857, 1995.

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

[37] H. Zhang and R. Liu. Mesh segmentation via recursive and visually salient spectral cuts. In *Proc. of Vision, Modeling, and Visualization*, pages 429–436, 2005.