3D-Facial Expression Synthesis and its Application to Face Recognition Systems

Leonel Ramírez-Valdez[1], Rogelio Hasimoto-Beltran[2]

[1,2]Centro de Investigación en Matemáticas(CIMAT)
Jalisco s/n, Col. Mineral de Valenciana, Guanajuato, Gto., México 36240
{lraval,hasimoto}@cimat,mx

ABSTRACT
One of the main problems in Face Recognition systems is the recognition of an input face with a different expression than the available in the training database. In this work, we propose a new 3D-face expression synthesis approach for expression independent face recognition systems (FRS). Different than current schemes in the literature, all the steps involved in our approach (face denoising, registration, and expression synthesis) are performed in the 3D domain. Our final goal is to increase the flexibility of 3D-FRS by allowing them to artificially generate multiple face expressions from a neutral expression face. A generic 3D-range image is modeled by the Finite Element Method with three simplified layers representing the skin, fatty tissue and the cranium. The face muscular anatomy is superimposed to the 3D model for the synthesis of expressions. Our approach can be divided into three main steps: **Denoising Algorithm**, which is applied to remove long peaks present in the original 3D-face samples; **Automatic Control Points Detection**, to detect particular facial landmarks such as eye and mouth corners, nose tip, etc., helpful in the recognition process; **Face Registration** of a 3D-face model with each sample face with neutral expression in the training database in order to augment its training set (with 18 predefined expressions). Additional expressions can be learned from input faces or an unknown expression can be transformed to the closest known expression. Our results show that the 3D-face model resembles perfectly the neutral expression faces in the training database while providing a natural change of expression. Moreover, the inclusion of our expression synthesis approach in a simple 3D-FRS based on Fisherfaces increased significantly the recognition rate without requiring complex 3D-face recognition schemes.

Keywords: Facial expression synthesis, Finite Element Method, feature points detection, eigenfaces, fisherfaces.

RESUMEN
Uno de los problemas principales en los sistemas de reconocimiento de caras es el reconocer una cara con una expresión distinta a la presente en la base de datos, esto es, son dependientes de la expresión de la cara de entrada. Con el propósito de flexibilizar los sistemas de reconocimiento de caras, se propone un método nuevo y eficiente para la síntesis de expresiones faciales en 3D y su aplicación a los sistemas de reconocimiento de caras independiente de la expresión (FRS). A diferencia de los métodos actuales en la literatura, todos los pasos involucrados en la síntesis de expresión facial (eliminación de ruido, registro y síntesis de expresión) son realizados en 3D. Nuestra meta es darle mayor flexibilización a los sistemas 3D-FRS para generar múltiples expresiones a partir de una cara base neutral, la cual es modelada con una malla de elemento finito de 3 capas que representan la piel, el tejido adiposo y el cráneo. Para la realización de la síntesis de expresiones en 3D, el modelo base es complementado con los músculos mas importantes que intervienen en la generación de expresiones faciales. El modelo propuesto se puede dividir en tres pasos principales: **Filtrado de Ruido**, usado para eliminar los picos (prominentes) presentes en las imágenes de profundidad; **Detección de Puntos de Control** en la base de datos de caras en 3D, como por ejemplo, punta y grosor de la nariz, puntos en los extremos de los ojos y de la boca, etc.; **Registro** del modelo base con cada una de las imágenes muestra con cara neutral en la base de datos de entrenamiento, para la generación de expresiones faciales sintéticas y su posterior inclusión en la base de datos misma para incrementar el conjunto de entrenamiento (a 18 expresiones predefinidas). Expresiones adicionales pueden ser aprendidas de las imágenes de entrada o bien expresiones desconocidas pueden ser transformadas a la expresión más cercana en la base de datos. Para medir la eficiencia del 3D-FRS con síntesis de expresiones, utilizamos una técnica muy simple en el reconocimiento de caras conocida con el nombre de FisherFace. Los resultados muestran que el método propuesto representa fielmente la imagen neutral de la base de datos y además, la adición de expresiones faciales sintéticas para el reconocimiento de caras efectivamente incrementa la taza de reconocimiento sin requerir algoritmos complejos para el reconocimiento de caras en 3D.

Palabras clave: Síntesis de expresiones faciales, Método de los elementos finitos, detección de los puntos de control, eigenfaces, fisherfaces.

1. Introduction

Recent advances in technology and increasing demand for applications such as biometric security systems, access control, virtual reality and human-computer interaction have created an enormous interest on automatic face recognition [32, 21, 1, 7]. In the past decades, most of the work was focused on using 2D gray-level or color images [32, 21]. However, it is well known that even small variations in illumination, pose and facial expression can drastically degrade the performance of the recognition system. 3D-face recognition has the potential to overcome these limitations since it relies purely on geometric features, which is not affected by lighting conditions. Additionally, 3D-face images can be rotated, allowing pose compensation variations. Certainly, there is a number of issues in 3D-face acquisition devices that need to be addressed, such as cost and low accuracy. It has been pointed out however [1, 7], that current technology is good enough for new developments in the 3D area.

Early work on 3D face recognition was done over a decade ago [1, 7]. Some of the reported algorithms use curvature analysis and feature extraction [13, 20, 23], which identify a number of facial features, including dimensions and curvatures, to perform recognition. Other methods use central or lateral profile information for surface matching [8, 2]. Appearance-based methods, such as Eigenfaces (PCA) and Fisherfaces (LDA) have been extended to 3D face recognition [31, 15, 16] and used with point clouds or range images. One important limitation to some existing 3D-face recognition approaches is the ability to handle expression changes. This problem is illustrated with the experiment described in [7]. Three galleries of 70 face images were built at different times, two with normal expression and one more with smiling expressions. Recognition was performed using PCA in 2D and 3D. Figure 1 shows the results of the experiment without expression change between gallery and probe; in both results 2D and 3D the recognition rate is above 90%, but when expression variation is introduced there is a noticeable drop in performance to 73% for 2D, and 55% for 3D. The relative degradation between 3D and 2D depends on the particular facial expression, but clearly, variation in facial expressions is a problem that must be addressed.

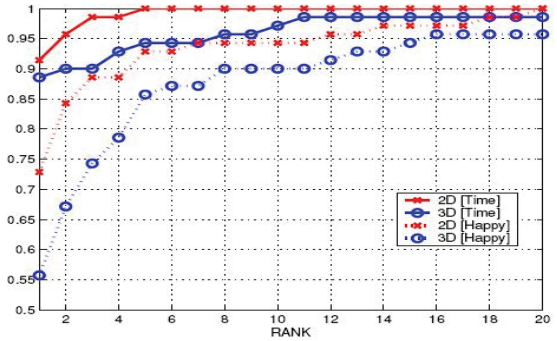


Figure 1. Effects of expression change on 3D and 2D recognition rates [7].

Chua et. al. [5] deal with facial expression changes identifying rigid areas of a 3D facial surface. Similarity between faces is computed comparing a set of unique point signatures. Experiments were conducted with a test set of 30 depth maps of 6 different people. Bronstein et. al. [34] use the assumption that the change of the geodesic distances due to facial expressions is insignificant, and proposed an expression-invariant representation for 3D face recognition. Moreno et al. [23] performed a 3D face recognition using segmentation based on Gaussian curvature to create a feature vector of the segmented regions. Using different expressions and poses Moreno, et. al. [24] achieved rank-one recognition rate of 78%. Lu et al. [22] used a 3D generic face model to create face images with synthesized expressions to construct an affine subspace for each subject; however the face synthesis and the recognition step was performed in 2D (with average recognition performance). In a recent paper, Lu and Jain [4] worked with deformable models using a hierarchical geodesic-based resampling

approach to extract landmarks for modeling facial surface deformations learned from a small group of subjects. The facial deformations are synthesized onto a user-specific 3D neutral model and recognition is based on 3D surface matching; they need natural facial expressions to synthesize artificial expression in the database neutral face samples. Heseltine et al. [15, 16] explore PCA and LDA based approaches with a variety of surface representations of three-dimensional facial structure; their gallery has six different samples per subject giving more chances to make a correct match instead of having a single sample per subject, but collecting 3D data of each subject with multiple expressions is not practical.

Anatomical 3D-face modeling and animation started in the 80's [28, 35], in order to provide realistic face expressions. These models incorporated the facial muscle structure [18-19-26-25-12], which was later incorporated with a facial action coding system [4] as a control procedure. These early models oversimplified the biomechanics of the face by considering an infinitesimal skin surface with no underlying structure. Newer schemes, on the other hand, are getting closer to the modeling of every single layer of the facial structure, such as epidermis, dermis, hypodermis, fatty tissue, muscles, skull and movable jaws [36]. Terzopoulus et.al. [28] stayed in the middle by considering a facial structure representation consisting of three layers, cutaneous tissue, subcutaneous tissue, and muscle layer. Regarding the physical model of the facial structure, the dominant models are the Mass-spring Damper (MSD) and the Finite Element Method (FEM).

In this work, we propose a new 3D-face system for the synthesis of expression considering the facial structure proposed in [28]. The difference of our facial structure is the use of the Finite Element Method instead of Mass-Spring Dumper (MSD) for the physical modeling of the face; additionally, our mesh consists of hexahedral rather than tetrahedral elements. Since the final application of our expression synthesis system is the 3D-face recognition, input facial samples are not subject to shear and twist forces; therefore, the synthesis of expression can be accurately represented by a hexahedral mesh. Our expression synthesis approach consists of a mesh denoising scheme, a feature point automatic selection scheme, and a 3D registration process between a standard model and an input face with neutral expression. Once the standard model is transformed into the input sample, synthetic expressions are generated in order to augment the 3D-face recognition database training set needed for the recognition process, as illustrated in Figure 2.

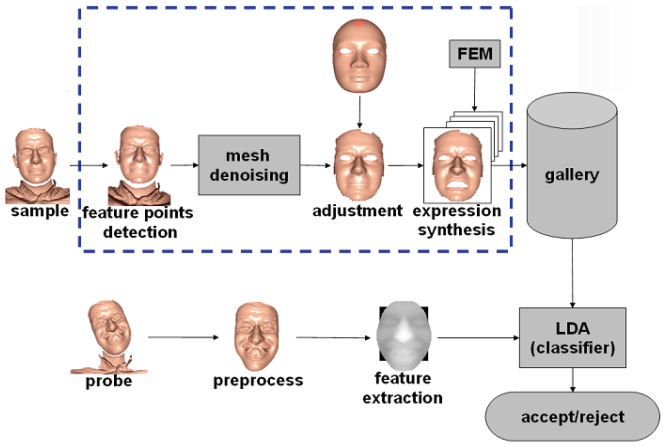


Figure 2. 3D face recognition system with the proposed gallery augmentation scheme (dotted square)

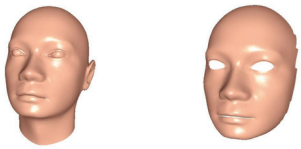
We would like to point out that we are not proposing a new scheme for 3D-face recognition; we are proposing a 3D facial expression synthesis, which can be used to increase significantly the 3D-face recognition rate without the need of having very complex face recognition schemes.

In the following section, we outline the process of generating synthesized expressions on the generic face with the Finite Element Method (FEM). The method for adjusting generic face to frontal faces of each subject is described in Section 2, including the proposed methods for feature points detection and mesh denoising. Face recognition using the fishersurface technique is described in Section 3. Experiments and results are presented in Section 4 and, finally, in Section 5 we give the conclusions and some ideas for future work.

2. Facial Expression Synthesis

Our proposed system considers a modified generic 3D-face model with facial structure and a 3D-face database with neutral expressions (sample faces). Before the expression synthesis step, the entire 3D-face database is subject to the following process: 3D-face sample denoising, automatic 3D feature-point selection, and 3D-registration between the generic model and each face sample. The generic model, which now has the appearance of the input face, is used to generate synthetic expressions (happiness, sadness, etc.) of the correspondence face sample to augment the training database. The above steps enclose our face expression synthesis system, as described in the following sections.

Generic model: We use a generic 3D-head model [26] consisting of 6054 four-sided polygons and 6,105 vertices (Figure 3(a)). Since we are only interested in the face area, polygons and vertices outside this area are eliminated, leaving 2676 polygons and 2777 vertices corresponding to the facial area (Figure 3(b)). In order to incorporate a physically-based approximation to facial tissue [28, 30] and produce more realistic facial expressions, we use a face mesh vertex unit normal multiplied by a factor to generate two additional layers under the original layer of vertices (one factor per layer). We have now 3 layers of vertices and 2 layers of hexahedrical elements, as shown in Figure 4. We provide distinct mechanical properties to each layer in order to simulate the facial structure. The lower layer acts as the cranium, the middle layer as fatty tissue and the top layer as skin.



(a) Original model (b) Edited model

Figure 3: Generic face model.

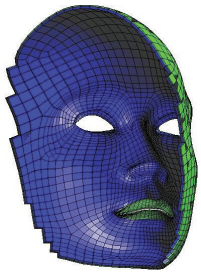
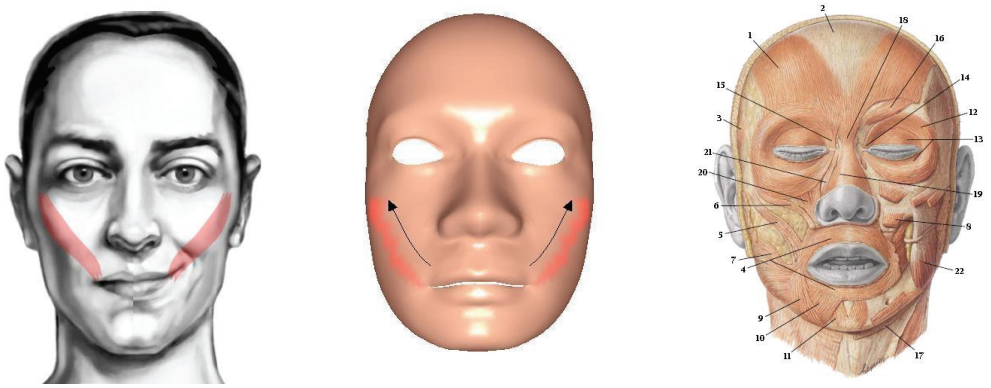


Figure 4. Facial layers

Facial muscles are defined as a set of vertices on which a force is exerted in a certain direction that depends on the location and form of each muscle. Based on [27, 10], we define 18 different muscles on the generic face model. Figure 5b shows an example of one muscle.

Facial Action Coding System: Once the physically-based face model has been set up, the facial expression simulation requires the mapping of a target expression into muscular activation. The Facial Action Coding System (FACS) [11] is the most widely used independent muscle control system to



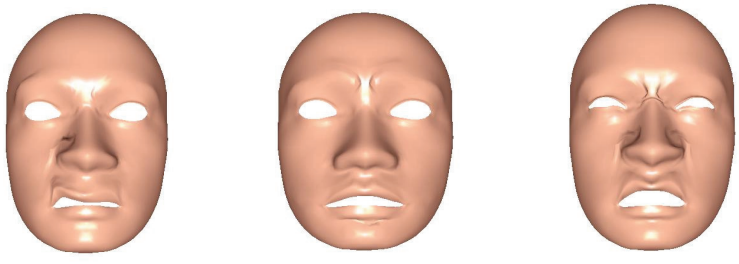
(a) Artnatomy illustration [10] (b) Face generic model (c) Facial muscles anatomy [27]

Figure 5. Muscle definition

describe and measure facial behavior. The system associates expression changes with the muscular actions and establishes a reliable mechanism to categorize facial expressions. FACS measurements are made in Action Units (AUs). A facial expression is defined by specific combinations of AUs, which provide just a descriptive score, without implications about the psychological or cultural meaning of the expression.

Our FEM implementation, which integrates a module from Botello, et al. [6], requires the following input data to solve the problem:

- **Material properties:** Based on experiments reported in [14], we give different material properties to each layer of hexahedrical elements in face generic model to simulate the facial layers. A Young's Modulus value of 5.6 kPa for the skin layer and 0.12 kPa for the fatty tissue. Quasi-incompressibility of tissues is modeled with a Poisson's ratio equal to 0.45 for both layers.
- **Prefixed nodes:** In order to simulate the cranial layer, nodes at the bottom layer are fixed (position does not change under external forces), same as border nodes for boundary conditions.
- **Point loads:** Depending on the target expression, we set the force parameters for each muscle and created 18 predefined expressions to be used for database augmentation. Figure 6 shows some of the generated facial expressions with FEM.



(a) Scorn (b) Fear (c) Pain

Figure 6. Examples of generated facial expressions

2.1 Input Face Pre-Processing Scheme

In order to be able to generate synthetic facial expressions to input face samples in the database, we transform (register) the generic face model into the corresponding neutral face sample. The generic 3D-face model resembles now the input face sample in the database with its corresponding muscular structure (facial muscles on the generic face are also deformed during the registration process). The registration process gives the generic face model the capability to generate different facial expressions of the input face.

Before registration, the face samples need to be preprocessed for noise elimination and feature-point detections. These schemes are described in the following subsections respectively.

2.1.1. Mesh Denoising

We propose a mesh denoising algorithm that combines both the Median and Laplacian filters [25]. The Laplacian filter provides good performance with short tailed noise distribution, whereas the Median filter works fine with long tailed noise distribution (outliers). Variations of the Laplacian filter has been widely used for mesh denoising [19, 29]. Its implementation for 3D meshes is very simple; the position of one vertex is just replaced with the average of the positions of the vertices inside its neighborhood. The Median filter on the other hand, takes the median of the neighborhood to obtain the new vertex coordinates.

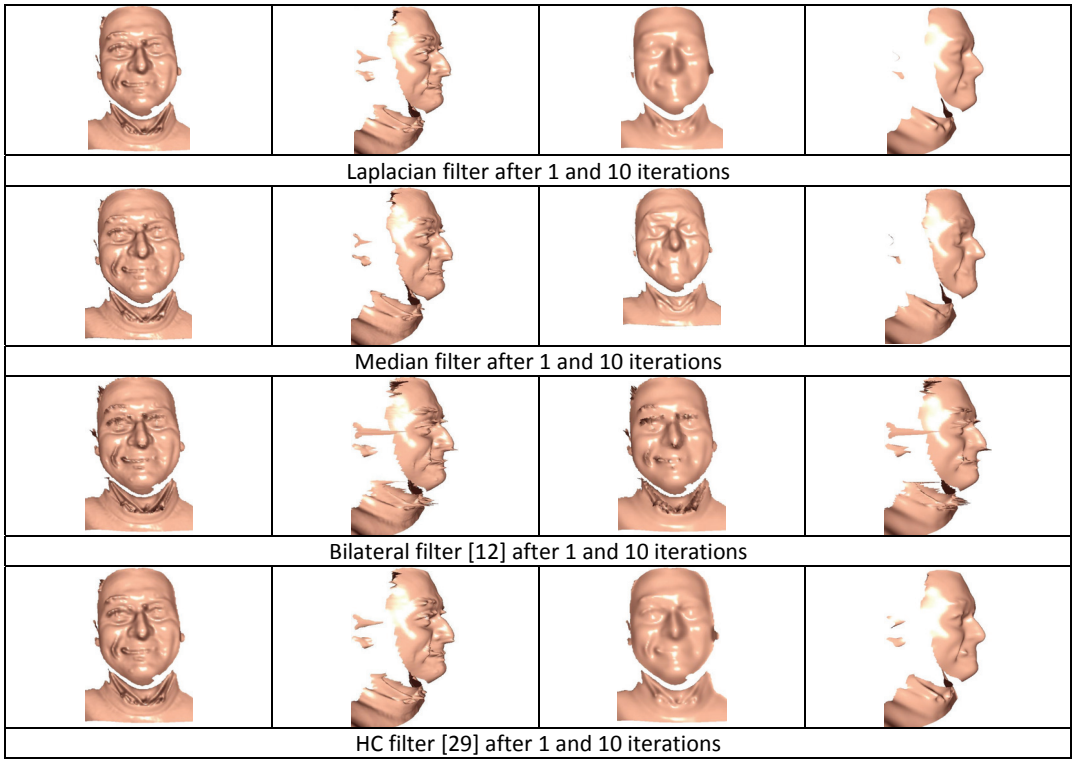
We take a combination of these two filters to create an iterative filtering algorithm, which keeps working until no substantial change is detected in the actual region of interest. Regions with different noise magnitudes have different number of iterations, that is, the proposed filter does not over smooth important features useful for the facial feature selection process. The basic idea of our algorithm consists on keeping the original position of vertices that have a little change between iterations, and update just the vertices with big changes. An extra copy of previous vertex position needs to be kept in memory for deciding updating the vertex position. The steps involved in the algorithm are

- Apply median filter to the mesh.
- Apply Laplacian filter to the mesh just filtered in the previous step.
- Compute the distance between the original positions and the new positions of vertices after the filtering.
- Define a threshold by multiplying the maximum absolute distance obtained in the previous step by a specified factor or percentage.
- Update the positions of those vertices where their distance calculated in step 3 is equal or greater than the defined threshold.
- Repeat the entire process until no vertices are updated or until k predefined iterations.

One of the problems with denoising algorithms is that, typically, they are not energy preserving, and Laplacian and median filter are not the exception, so in 3D meshes, they will shrink the mesh in order to be represented optimally. If the number of iterations *k* goes to infinity, the mesh will be shrinking towards a single point. Our algorithm shrinks the mesh as well, but compared to other approaches reported in literature, it just requires two very intuitive parameters. One parameter is the number of iterations *k*, which is common in the most of the algorithms; this number influences directly the elimination of long peaks, however, if it exaggerates the number of iterations, a strong shrinkage and over smoothing will appear. The second parameter is the change percentage *w*, which is related to the vertex with

the biggest displacement in the current iteration. It defines the threshold for which the algorithm applies the filtering scheme on a specified region. For *w = 0.8* (big), only the vertices with displacement equal or greater than 80% with respect to the biggest displacement are modified; peaks disappear gradually without damaging natural sharp regions. For small values of *w* (*<0.2)*, the number of iterations increases and may produce over smooth regions. For the purpose of this work and for the particular noise present in our 3D-face database, we experimentally found (only few images were damaged by the presence of noise) that *w~0.8* with 15 iterations provides the best visually noise reduction. Large noise peaks are practically removed during the first 10 iterations, and approximately 5 additional iterations are needed to eliminate shorter and smoother peaks without affecting important edge information in the 3D face image.

Table 1 shows the results of the proposed algorithm (last row), it can be seen that after 20 iterations, long peaks on facial surface are gone while still preserving sharp features. For comparison purposes, we apply different algorithms reported in literature to the same noisy facial surface, we can see that this algorithms have a good performance with short tailed noise, but with long peaks (which is the common case) our method (last row) presents much better results. It is difficult to quantify the effectiveness of our filter (in terms of the Mean Square Error-MSE for example) since we do not have the original image to compare with (same expression without noise); therefore, the validation process is performed visually. There may be a possible solution to quantify the noise reduction effectiveness of our scheme (although time consuming) by taking a clean image from the same face set and manually select the control points and muscle transformation to convert it into the noise image; this would be a very tedious task in the case of a statistical analysis of the filter.



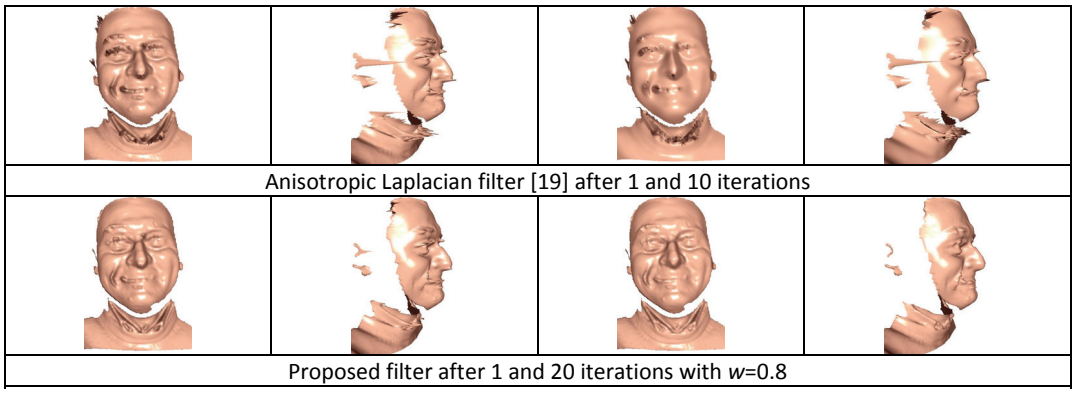


Table 1. Mesh denoising with different algorithms

2.1.2 *Feature points detection*

For feature point detections on 3D facial surfaces, we proposed a heuristic algorithm based on surface curvature (Figure 7) and statistical distance information between facial components (Table 2). The algorithm is similar to the one presented in [9], with the following differences; we do not use 2D color information, our scheme requires 3D-frontal faces with neutral expressions as input, and we developed a new heuristic scheme which first tries to localize those regions where is more likely to find face features such as the nose tip, mouth corners, eyes corners, etc. (see Table 2). Our feature point detections scheme consists of the following steps:

1. **Facial Segmentation:** This step makes use of the Shape Index value *(avg(S))*, which represents a pose invariant representation of the face surface curvature [9]. Experimentally, we found that the facial region has an average Shape Index value (avg(S)) greater than 0.7, as shown by the yellow and red regions in Figure

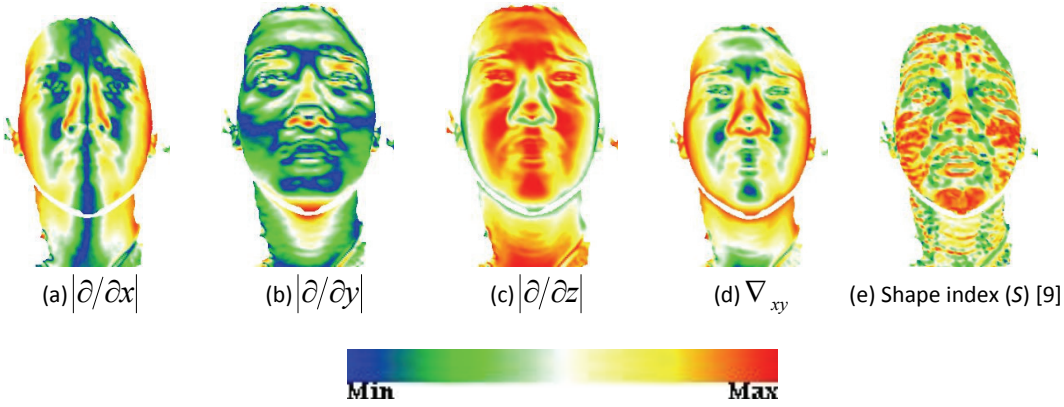


Figure 7. surface descriptors

Point A	Point B	Direction	Minimum (mm)	Average (mm)	Maximum (mm)
Top of Head	Nose Tip	Vertical	84.0	121.7	181.0
Nose Tip	Mouth	Vertical	16.9	34.4	46.3
Nose Tip	Chin	Vertical	49.7	70.6	95.4
Nose Tip	Nose Bridge	Vertical	21.2	32.2	48.4
Nose Tip	Inside Eye	Horizontal	13.2	17.5	23.6
Inside Eye	Outside Eye	Horizontal	15.9	30.6	39.7

Table 2. Statistical model of the inter-anchor point distances on a face. Each set of distances are calculated from point A to point B only in the specified direction [9].

7(e). The first step of the algorithm is to filter out face regions with low S<0.375 and $|\partial/\partial z|$ <0.5". We average the survival regions and filter out again those regions for which avg(S)≤0.7, as illustrated in Figure 8.

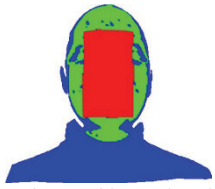
2. **Nose tip detection:** Nose tip is not necessarily the highest point in Z direction. To obtain this point we take extreme values of regions with ∇_{xy} <0.3 in X and Y direction to construct a bounding box (Figure 9(a)). Within this bounding box, we search for the biggest region where $|\partial/\partial z|$ ≤0.5, which is characteristic of nasal corners. Then, we look for extreme values in Y direction of nasal corners and reduce the bounding box, where we search for the region with the highest point in Z direction with ∇_{xy} <0.25. The nose tip will be the centroid of this region (Figure 9(b)).

(a) Blue: *S*<0.375 and $|\partial/\partial z|$ <0.5

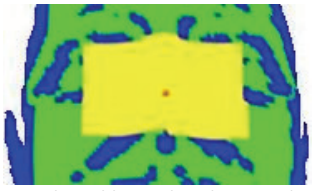
(b) Green: *avg(S)*>0.7

(c) Green: Segmented facial region

Figure 8. Facial segmentation



(a) Red: Initial bounding box

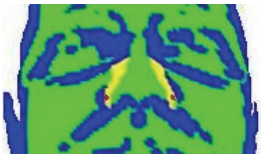


(b) Yellow: Reduced bounding box. Red: Nose tip

Figure 9. Nose tip detection

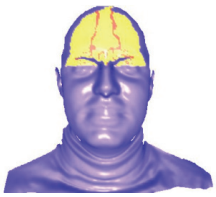
3. **Inner eye corners:** Inner eye corners are regions with small Shape Index, as shown in Figure 7(e). In order to accelerate their detection, we use statistical information between the nose tip and nose bridge (Table 2) to construct a bounding box where we search the two biggest regions with $S \leq 0.25$. The centroid of each region will be the respective inner eye corner.
4. **Outer eye corners:** We apply the same idea as in the last step constructing two bounding boxes with the distance between inner and outer eye corners (Table 2) and looking for the biggest regions with $S<0.5$, $|\partial/\partial x| \leq 0.5$ and $\nabla_{xy} < 0.25$. The centroid of each region will be the respective outer eye corner.
5. **Nose corners:** Nose corners are regions with $|\partial/\partial z| \leq 0.5$ and $|\partial/\partial x| > 0.65$, as shown in Figure 10. We construct a bounding box with the outer eye corners in X direction and in Y direction with the half of the distance between nose tip and inner eye corners. Nose corners will be the maximum and minimum values in X direction of the respective region.
6. **Mouth corners:** We construct a bounding box with the distance between nose tip and mouth in Y direction and with the outer eye corners in X direction. Inside of this box, we search for regions where $S<0.75$ and $\nabla_{xy} < 0.35$, which characterize lips. Taking extreme values of these regions in Y direction we reduce the initial bounding box. In this new bounding box, mouth corners will be regions characterized by $S<0.25$.
7. **Cheek borders:** Cheek borders are regions with $|\partial/\partial z| \leq 0.4$ (Figure 7(c)). We search for these regions inside two bounding boxes constructed with mouth and eyes in Y direction, and outer eye corners in X direction, one box for each side, respectively. The centroid of the biggest regions will be taken as cheek borders.
8. **Forehead border:** To detect this point, we search for regions located above the eyes with $0.19 < |\partial/\partial x| < 0.25$ (Figure 11(a)), the upper border will be delimited by $|\partial/\partial z| > 0.6$. We take the two longest regions in Y direction and the upper value of any region will be the forehead border (Figure 11(b)).
9. **Chin border:** Finally, we construct a bounding box with the mouth corners and the bottom facial region detected in step 1. Inside this box we search for regions where $S>0.85$ and

a) Yellow: Bounding box. Red: $|\partial/\partial z| \leq 0.5$ and $|\partial/\partial x| > 0.65$



(b) Yellow: Biggest regions. Red: Nose corners detected

Figure 10. Nose corners detection



a) Yellow: Bounding box. Red: $0.19 < |\partial/\partial x| < 0.25$

b) Yellow: Biggest regions. Red: Forehead border detected

Figure 11. Forehead border detection

∇_{xy} <0.3, which characterize chin (Figure 7(e) and (d)). Then we get the biggest region and the minimum value in Y direction will be the chin border.

Figure 12 shows three examples of feature points detected by the proposed algorithm using the two frontal meshes of the GavabDB database [24] (for a total of 122 samples). With the right input samples, that is front neutral expression, no occlusion, and continues mesh (without holes); the percentage of locating all feature points correctly is 93.5%. The success rate reported in [9] is 99.1%, but they considered a success when 3 of 5 feature points regions were correctly detected. Under the same success reference, we obtained a 100% of feature point detection.

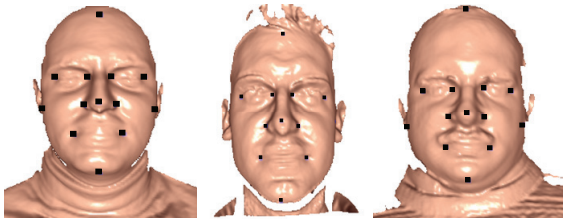


Figure 12. Automatically detected feature points

2.1.3 Face Registration

Once we have detected the feature points on both the generic face model (marked manually) and the input 3D faces (detected automatically), we proceed to transform (register) the generic face model into the input face by using their corresponding feature points. In order to keep track of the generic face model changes, as well as its muscular structure connectivity during the registration process, we divide the face registration process into three steps: *global alignment*, *deformation in XY plane*, and *deformation in Z* direction.

Global alignment consists of a rigid transformation that minimizes the distance between corresponding feature points on both faces (input face and generic face model). To find the best rigid transformation, we borrow the scheme of Unit Quaternion developed by Horn [17]. After the global alignment, the generic face model is deformed in such a way that all feature points in both faces become exactly aligned, and all vertices affected by this deformation are linearly mapped. We then applied a simple linear mapping in the horizontal and vertical directions as shown in Figure 13.

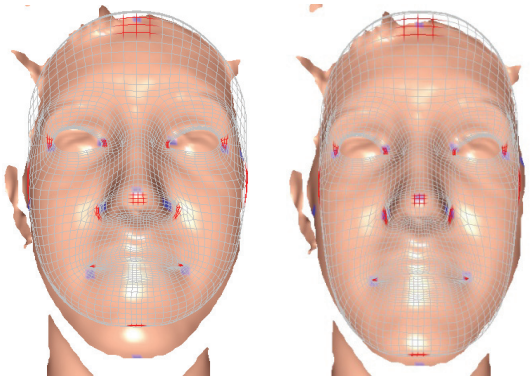


Figure 13. Registration process of the generic face model (wired mesh with red feature points) and input face (blue feature points) in the XY plane

Finally, for the deformation in Z direction, we cylindrically project the generic facial mesh and locate each vertex inside one of the polygons of the input facial mesh. We obtain the position of each vertex by barycentric coordinates in the respective polygon. After the registration process, the generic face model can be used for the generation of synthetic facial expressions, as shown in Figure 14. The performance of our synthetic expression scheme is analyzed based on how similar or realistic the synthetic expressions are compared to the closest real face expressions in the database. Smooth face expressions, in which the mouth is barely open such as smile, fear, determination, doubt, etc., are accurately represented (very realistic expression); on the other hand, over-reacting expressions with wide-open mouth are not well represented because of the limitation of our skin-cranium model. We use

Input face	Register face (generic model)	Anger	Fear
			

Figure 14: Examples of generated facial expressions

a simplified model where the skin-layer width is constant all over the face surface and localized at a constant distance from the cranium layer (this is not true particularly around the mouth and cheek areas). As we will see in Section 3, this simplified model allow us to create synthetic face expressions that fall in the subspace of the expression we are trying to synthesize, which is relevant in face recognition applications. In fact, we use face recognition to indirectly prove the performance of our face synthesis scheme under overreacting face expressions.

3. Application of Expression Synthesis to Face Recognition Systems

In order to validate our expression synthesis approach in a real case, it is applied in a 3D-face recognition system. As stated before, we do not propose a new face recognition scheme; our only objective here is to show that a good expression synthesis approach with a baseline face recognition scheme (simple face recognition scheme) draws similar results than other complex state of the art schemes [4, 23].

Since we are increasing the training set in the database by expression synthesis, it is natural to think about a face recognition method with good class discriminating power, such as Fisherfaces (LDA) [16, 3]. The LDA method maximizes class separation and minimizes variation between multiple faces of the same subject. Consider our facial surfaces as range images, represented as a vector of length 9,891. To create this range images, we fit a rectangular mesh with 9,891 vertices to the facial surfaces, with the nose tip on the center and moving only the Z coordinate of each vertex, so every facial surface has the same number of vertices. To avoid possible peripheral noise an elliptical mask is used to crop range images and holes inside the ellipse are filled with the average neighborhood information.

Suppose we have the training set partitioned in c classes in our 9,891 dimensional space, where each single class X_n has 19 (18 synthetic expressions plus 1 neutral) surface vectors Γ_{ni} of the same person. We need to obtain the projection matrix:

$$U_{opt} = U_{lda} U_{pca} \tag{1}$$

where

$$U_{lda} = \max_{U} \left(\frac{\left| U^T U_{pca}^T S_B U_{pca} U \right|}{\left| U^T U_{pca}^T S_w U_{pca} U \right|} \right) \tag{2}$$

$$U_{pca} = \max_{U} \left(\left| U^T S_T U \right| \right) \tag{3}$$

$$S_T = \sum_{n=1}^{c} \sum_{i=1}^{|X_n|} (\Gamma_{ni} - \Psi)(\Gamma_{ni} - \Psi)^T \tag{4}$$

$$S_B = \sum_{n=1}^{c} |X_n| (\Psi_n - \Psi)(\Psi_n - \Psi)^T \quad (5)$$

$$S_w = \sum_{n=1}^{c} \sum_{i=1}^{|X_n|} (\Gamma_{ni} - \Psi_n)(\Gamma_{ni} - \Psi_n)^T \quad (6)$$

$$\Psi = \frac{1}{\sum_{m=1}^{c} |X_m|} \sum_{n=1}^{c} \sum_{i=1}^{|X_n|} \Gamma_{ni} \quad (7)$$

$$\Psi_n = \frac{1}{|X_n|} \sum_{i=1}^{|X_n|} \Gamma_{ni} \quad (8)$$

Once the projection matrix has been calculated, facial surfaces are projected into a space of *c-1* dimensions by a simple matrix multiplication:

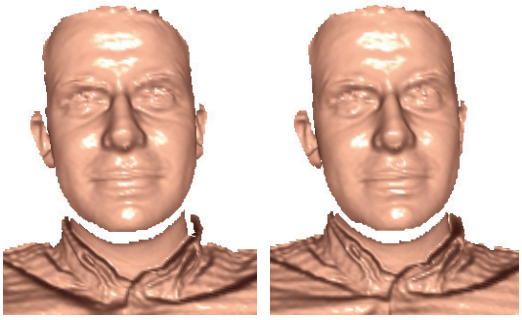
$$\Omega = (\Gamma - \Psi)^T U_{opt} \quad (9)$$

The resulting vector *Ω* represents the facial structure which is compared using any distance measure to determine similarity of facial surfaces. Notice that U_{pca} is used to ensure non-singularity reducing dimensionality of the within class (S_w) and between class (S_b) scatter matrices to *M-c* where *M* is the size of the training set.

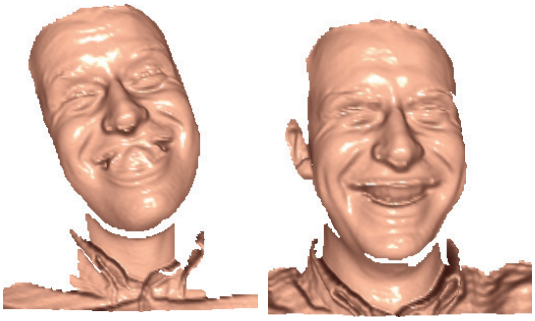
4. Experiments and Results

Using the GavabDB 3D facial database [24], we created two sets, one set is for the recognition step and the other for the training step. There are five frontal facial surfaces per subject, two with neutral expression and three with expression variations as shown in Figure 15.

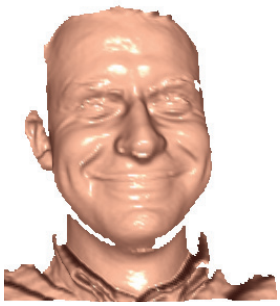
For the training set, we processed one of the neutral expression samples, and the four remaining samples were used for the recognition step. A total of 18 synthesized facial surfaces per subject were created from the original neutral expression, having a total of 19 samples per subject (1 neutral + 18 synthesized) out of 57 subjects. We have a total of 228 (57x4) sample faces for the recognition step.



a) Neutral b) Neutral 2



c) Gesture d) Laugh



e) Smile

Figure 15. Face sample set in GavabDB database [48].

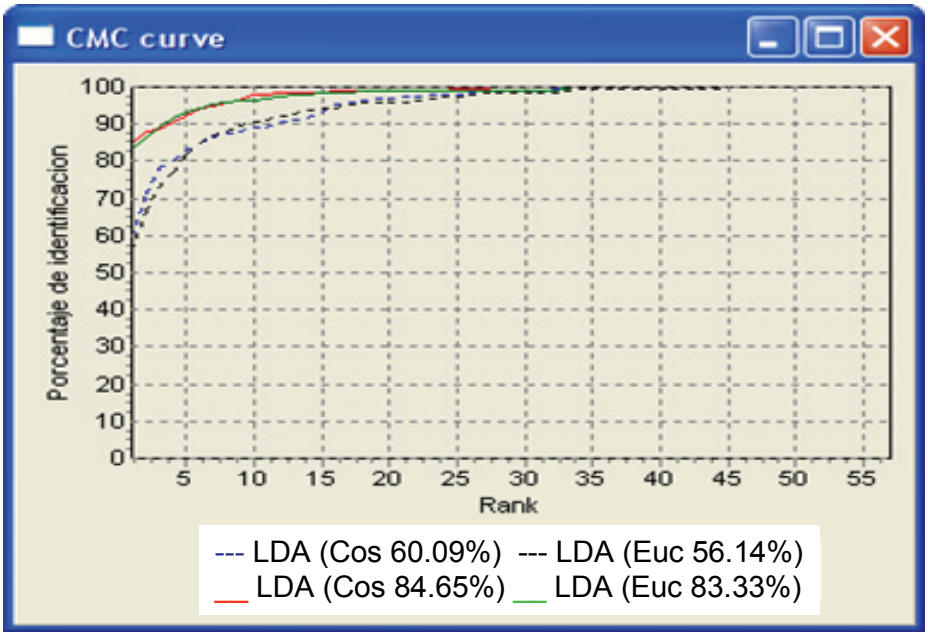


Figure 16. Recognition rate with (continues lines) and without (dashed lines) expression synthesis augmentation

In the first experiment, we applied Fisherfaces (LDA) [3, 15, 3] to the training set using only the neutral facial surface per subject, that is without expression synthesis. In the second experiment, we added the synthesized facial surfaces to the training set and applied again the Fisherfaces method. Results are shown in Figure 16 as Cumulative Match Characteristics (CMC) curves, using cosine and Euclidian distance metrics. It can be appreciated that the recognition rate improves substantially (~20%) in rank-1. In rank-5, the CMC is approximately 94%. Compared to [4], we are ~2-3% below in rank-1 and approximately at the same level in rank-5, by only using the LDA scheme. The difference between [4] and our model is complexity; we are using the simplest general scheme for face recognition, while [4] uses more complex Hierarchical Facial Surface Sampling (detection of landmarks), Deformation Transfer and Synthesis (registration, mapping of neutral to non-neutral expressions and interpolation tools), and a creation of Deformable Models. It is difficult to make comparisons among different face recognition systems in the literature because they use different database. Problems still remain even when using the same database because some schemes may not use the entire database or may filter out anomalies in the 3D faces that alter the results. We can closely compare to the scheme reported in [23], even though we use only 57 different people out of the 61. They report a recognition rate of 78% for frontal samples with neutral expression, and a 68% recognition rate for samples with non-neutral expression. The extended GavabDB 3D facial database with our face synthesis scheme has a recognition rate of 84.65%, again using the simplest face recognition scheme based on LDA.

5. Conclusions and Future Work

We have developed an efficient scheme for the 3D-synthesis of expressions from neutral face samples. The main idea is to create expression independent 3D-face recognition systems by adding new face expressions in the training

database. To get a synthesized expression from neutral 3D-face samples (in the training database), we proposed a simple mesh denoising scheme, an effective feature points detection algorithm, and efficient registration process between a generic face model an the input face sample. Our scheme, which is based only on 3D information, improves the percentage of recognition by ~20%, staying very close to more complex schemes in the literature.

References

[1] L. Akarun, B. Gökberk, A. Salah. 3D Face Recognition for Biometric Applications. 13th European Signal Processing Conference (EUSIPCO), September 2005.

[2] C. Beumier, M. Acheroy. Automatic 3D Face Authentication. Image and Vision Computing, 18(4):315-321, 2000.

[3] P. Belhumeur, J. Hespanha, D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):711-720, 1997.

[4] X. Lu, A. Jain. Deformation Modeling for Robust 3D Face Matching. IEEE Trans. Pattern Analysis and Machine Intelligence, 2007.

[5] C. Chua, F. Han, T. Ho. 3D Human Face Recognition Using Point Signature. Proc. IEEE International Conference on Automatic Face and Gesture Recognition , 233-238, 2000.

[6] S. Botello, H. Esqueda, F. Gómez, M. Moreles, E. Oñate. Finite Element Method and Applications. Monografía M-AUGTO-2, Guanajuato, Gto., Noviembre 2004.

[7] K. Bowyer, K. Chang, P. Flynn. A Survey of Approaches and Challenges in 3D and Multi-modal 3D+2D Face Recognition. Notre Dame Computer Science and Engineering Technical Report, 2004.

[8] J. Cartoux, J. LaPreste, M. Richetin. Face Authentication or Recognition by Profile Extraction from Image Ranges. Proceedings of the Workshop on Interpretation of 3D Scenes, 194-199, 1989.

[9] D. Colbry, G. Stockman, A. Jain. Detection of Anchor Points for 3D Face Verification. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005.

[10] V. Contreras. Artnatomy. Anatomical Basis of Facial Expression Learning Tool. Spain, 2005. Available at http://www.artnatomia.net.

[11] P. Ekman, W. Friesen. Manual for the Facial Action Coding System. Consulting Psychologists Press, Palo Alto, 1977.

[12] S. Fleishman, I. Drori, D. Cohen-Or. Bilateral Mesh Denoising. International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2003, 22(3):950-953, 2003.

[13] G. Gordon. Face Recognition based on Depth and Curvature Features. 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92), 108-110, 1992.

[14] F. Hendriks, D. Brokken, J. van Eemeren, C. Oomens, F. Baaijens, J. Horsten. A Numerical-Experimental Method to Characterize the Non-Linear Mechanical Behaviour of Human Skin. Skin Research and Technology 9(3):274-283, 2003.

[15] T. Heseltine, N. Pears, J. Austin. Three-Dimensional Face Recognition: An Eigensurface Approach. International Conference on Image Processing (ICIP'04), 2:1421-1424, 2004.

[16] T. Heseltine, N. Pears, J. Austin. Three-Dimensional Face Recognition: A Fishersurface Approach. Proceedings of the International Conference on Image Analysis and Recognition, LNCS 3212:684-691, 2004.

[17] B. Horn. Closed-Form Solution of Absolute Orientation using Unit Quaternions. Journal of the Optical Society of America, 4(4):629-642 1987.

[18] J. Huang, V. Blanz, and B. Heisele. Face Recognition with Support Vector Machines and 3D Head Models. International Workshop on Pattern Recognition with Support Vector Machines (SVM2002), Niagara Falls, Canada, 334-341, 2002.

[19] H. Huang, U. Ascher. Fast Denoising of Surface Meshes with Intrinsic Texture. Preprint, 2006.

[20] J. Lee, E. Milios. Matching Range Images of Human Faces. Third International Conference on Computer Vision, 722-726, 1990.

[21] S. Li, A. Jain (Editors). Handbook of Face Recognition. Springer, 2005.

[22] X. Lu, R. Hsu, A. Jain, B. Kamgar-Parsi, B. Kamgar-Parsi. Face Recognition with 3D Model-Based Synthesis. International Conference on Biometric Authentication (ICBA'04), LNCS 3072:139-146, 2004.

[23] A. Moreno, A. Sánchez, J. Vélez, F. Díaz. Face Recognition using 3D Surface- Extracted Descriptors. Iris Machine Vision and Image Processing Conference (IMVIPC 2003), September 2003.

[24] A. Moreno, A. Sanchez. GavabDB: A 3D Face Database. Proceedings of the Second COST275 Workshop on Biometrics on the Internet, Vigo (Spain), 2004.

[25] H. Myler, A. Weeks. The pocket handbook of image processing algorithms in C, Prentice Hall, 1993.

[26] Singular Inversions Inc., FaceGen Modeller. Available at http://www.facegen.com/downloads.htm.

[27] J. Sobotta. Atlas of Human Anatomy. Volume 1: Head, Neck, Upper Limb. Lippincot Williams & Wilkins, 13th english/english edition, 2001.

[28] D. Terzopoulos, K. Waters. Physically-Based Facial Modeling, Analysis and Animation. Journal of Visualization and Computer Animation, 1:73-80, 1990.

[29] J. Vollmer, R. Mencl, H. Müller. Improved Laplacian Smoothing of Noisy Surface Meshes. Computer Graphics Forum, 18(3):131-138, 1999.

[30] K.Waters. A Muscle Model for Animating Three-Dimensional Facial Expression. 14th Annual Conference on Computer Graphics and Interactive Techniques, 17-24, 1987.

[31] C. Xu, Y. Wang, T. Tan, L, Quan. A New Attempt to Face Recognition using 3D Eigenfaces. 6th. Asian Conference on Computer Vision, 2:884-889, 2004.

[32] W. Zao, R. Chellappa, P. Phillips, A. Rosenfeld. Face Recognition: A Literature Survey. ACM Computing Surveys, 35(4):399-458, 2003.

[33] O. Zienkiewicz, R. Taylor. The Finite Element Method. Volume 1: The Basis. Butterworth-Heinemann, 5th. Edition, 2000.

[34] A. M. Bronstein, M. M. Bronstein, R. Kimmel. Three-Dimensional Face Recognition. International Journal of Computer Vision, 64(1):5-30, 2005.

[35] F. I. Parke. "SIGGRAPH'89 Course Notes Vol 22: State of the Art Facial Animation", July, 1989.

[36] Y. Zhang, E. C. Prakash, E. Sung. Modeling and Animation of Individualized Faces for 3D Facial Expression Synthesis. International Journal of Imaging Systems and Technology , 13(1):42-64, 2003.

Authors' Biography

Rogelio HASIMOTO-BELTRÁN

Dr. Hasimoto received his Ph.D. degree in electrical and computer engineering from the University of Delaware, USA, in 2001. From 2000-2002, he served as a senior software engineer at Akamai Technologies in San Mateo, CA, USA. In 2003, Dr. Hasimoto joined the Department of Computer Sciences at the Center for Research in Mathematics (CIMAT) in Guanajuato, Mexico. His research interests include image/video processing and compression, robust multimedia transmission, biometric systems and multimedia encryption. He is a member of the IEEE association and the National Researchers System (SNI).

Leonel RAMÍREZ-VALDEZ

He received the B.S. degree in computer science in 2002 from the Universidad de Guadalajara, Guadalajara, Jalisco, Mexico and the M.Sc. degree in computer science and industrial mathematics from the Center for Research in Mathematics (CIMAT), Guanajuato, Gto., Mexico in 2007. His research interests include software development, database systems, image processing and computer vision. Currently, he is a software engineer at the Software Development Department of CIMAT, and he is developing augmented reality and real-time video processing applications.