3D facial animation based on structural registration

Andréa Britto Mattos, Jesús P. Mena-Chalco and Roberto Marcondes Cesar Jr.
Institute of Mathematics and Statistics - University of São Paulo
{*dedea,jmena,roberto.cesar*}*@vision.ime.usp.br*

***Abstract*—Our work[1] describes a facial animation method that uses real three-dimensional models of people, acquired by a 3D scanner. The models are animated using two techniques: applying a linear interpolation process using scanned expressions as keyframes and generating new facial expressions from a neutral face, using a parametric model based on the MPEG-4 animation standard. In both cases, point-to-point correspondences among the facial meshes are required, in order to register the interpolated meshes and to adapt the neutral mesh to a parametric deformation model. We propose an alternative to the dense correspondence computation by introducing the idea of selecting a sparse set of corresponding points and setting an initial triangulation refined through a subdivision process that matches the intermediate points. Our approach uses structural graph matching to automatically detect the initial set of points, given a 3D face in which the landmarks are previously selected. The proposed method can be applied to a variety of problems, enabling facial expression transfer between different model types, animation of reconstructed faces from 2D photographs and automatic caricature generation. Also, our approach can be used to solve other problems related to computer facial manipulation.**

***Keywords*-facial animation; three-dimensional registration; structural methods;**

I. Introduction

Computer facial animation has always attracted much interest among researchers, being widely studied due to its many applications. Our study explores facial animation using real three-dimensional models of people, obtained by a 3D scanner. We propose different animation strategies based on a new technique for matching corresponding points on facial meshes.

Facial animation methods can consider facial keyposes or estimate new expressions from a neutral face. For the first case, the interpolation method is commonly applied, using different facial expressions as keyframes. This strategy generates smooth transitions by the computation of the intermediate points between two poses. For the second approach, a parametrization model is often used, modifying the neutral face through a set of parameters applied to a group of feature points. Previous studies have widely adopted the international parametric model proposed by the MPEG-4 animation standard to perform this task. In our study, we discuss the implementation of this two animation strategies.

For the interpolation method, we use models displaying different facial expressions, acquired by the scanner. It is possible to obtain a simple interpolation when the meshes to be animated have the same topology, i.e., for each vertex of one mesh, the corresponding vertex of the other mesh is known, so there is a matching between all the points. However, this restriction is not satisfied when dealing with scanned data. Similarly, for the parametric animation, there must be a mapping between the mesh points and the parametric model that defines its deformation on the application of each parameter. Thus, this approach also requires, as an initial step, the definition of correspondences between face models.

Previous studies have already investigated strategies for computing one-to-one correspondences in facial meshes. The research line followed by [1], [2], [3], [4], [5] uses an optical flow based algorithm, while others rely on the approach of manually selecting sparse points on one mesh, deforming it according to this points, and projecting the deformed model on the target mesh [6], [7], [8], [9]. Some of these methods are not able to register faces with different facial expressions and others require a large training dataset, with a variety of facial expressions. Moreover, all these approaches have in common the focus of dense correspondence searching, which can lead to some problems, such as insufficient number of matches or bad matches in plain facial regions, such as forehead and cheeks, where there is little geometry and texture variation.

Also, most of these methods require the manual selection of a set of corresponding points on the facial meshes, which can be an exhaustive process and prone to errors. Previous works have treated methods for automatic facial landmarking, and the *Active Appearance Model* and *Active Shape Model* methods are widely used, especially in 2D images. However, these algorithms require a manually labeled training set and have poor performance when considering facial features that are not in the training database, implying a lack of local robustness.

Other methods are also used to perform the landmarking procedure [10], [11], based on the approach of aligning the mesh and a landmark template, using deformations and projections. As the similar matching approach, these methods are unable, for example, to detect landmarks on different expressions. On the other hand, the studies described on [12], [13], though considering a small landmark set (eyes and mouth corners and nose tip), showed that the idea of using structural information can provide good and robust results.

A. Contributions

Given the previous description of open problems, the main goals of our work are:

Avoid the disadvantages of the dense correspondence search among facial meshes: In our approach, we only establish a few corresponding points that form an sparse model,

[1]This paper gives a brief overview for the Master Thesis of the first author. The full text, videos and additional material are available at www.vision.ime.usp.br/~dedea/research.

and the matching of intermediate regions is performed by a subdivision process. Unlike some of the previous methods, this strategy is able to register models with different expressions.

Automatically find the initial points set: Most of the mentioned studies use manual landmarking to assist the facial registration process. We propose a new alternative to automatically match a set of points in different face meshes, discarding the need of a large training set, by introducing a novel structural approach for 3D landmarking.

Animate scanned faces using linear interpolation: Given the registration output, we use the interpolation method to obtain an animation in which the keyposes refer to different facial expressions.

Propose a parametric MPEG-4 based animation method: We show how the facial landmarking and subdivision procedures can be used to define the mesh deformation required on parametric animations.

This document is organized as follows. Section II gives a brief description of the used database and section III describes how the scanned faces were registered, using the proposed structural landmarking and subdivision strategies. Then, sections IV and V describe how the registration method allows facial animation, using linear interpolation and parametric model, respectively. The work is concluded in section VI.

II. Database

The facial database used in this work was obtained through a 3D scanner as described in [14]. Each acquisition is composed of registered geometry and texture data, i.e., each 3D vertex is associated with a point in the texture image.

The neutral expression and the six Ekman's facial expressions (joy, sadness, surprise, anger, disgust and fear) were captured for different individuals, generating a set of independent raw meshes that does not have any type of correspondence. A total of 30 individuals were considered, displaying the seven mentioned expressions, yielding 210 different meshes that were used as input to the matching and animation algorithms that will be described in the subsequent sections.

III. Structural Registration

A. Facial Landmark Matching

This section describes a novel facial landmarking approach that is based on point matching between facial meshes and how the facial registration is obtained from this set of landmarks.

Our landmarking method is based on structural graph matching and consists of a 3D extension for the 2D approach introduced by [15]. The structural methods consider an image as a set of related parts represented by a graph, whose vertices relate to the image parts and edges represent the structural relations between them. These graphs can be used so that one or more images can be compared by finding matches between the graphs that represent them[2].

In our approach, we manually select landmarks in one mesh (called *model*) and the corresponding points are automatically matched to other meshes (called *inputs*), that may refer to different individuals or different expressions. We use 48 landmarks that form an initial model G_m (Figure 1).

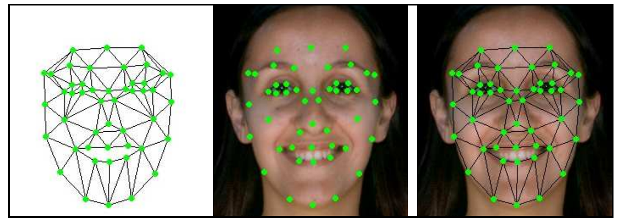


Figure 1. Model graph G_m, build from a set of 48 facial landmarks.

Then, a set of points is automatically selected on each input, using the *good features to track* detector [16]. These points represent the candidate vertices of the input G_i. Our goal is to assign an isomorphism[3] between the G_m vertices and a subset of the G_i vertices, in order to maximize an optimization function. Figure 2 shows an example of one model (with manually selected points), one input (with automatically detected candidates) and a simplified isomorphism relation scheme.

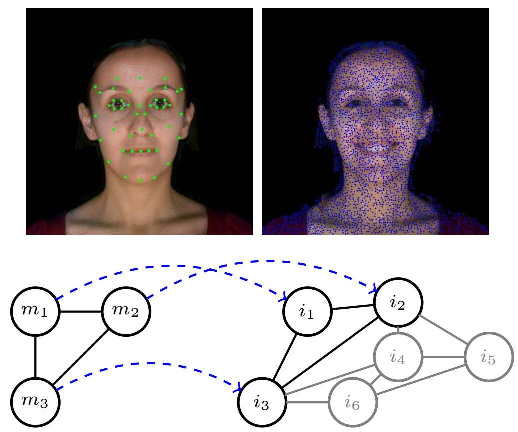


Figure 2. Point matching scheme between model (left) and input (right).

The correspondence between the model's points and input's candidates is computed using a greedy strategy. We assign costs to the matching of each model point and every input candidate and select the lowest cost matching. The structural approach requires the definition of costs to the graph's vertices and edges. The vertex cost consider the point's local information and the edge cost takes into account the graph structure. Formally, for each pair (v_m, v_i), where v_m corresponds to the model's vertices, and v_i corresponds to the input's vertices, we aim to minimize the following function:

$$E(v_m, v_i) = \lambda d_A(v_m, v_i) + (1 - \lambda) d_S(v_m, v_i) \qquad (1)$$

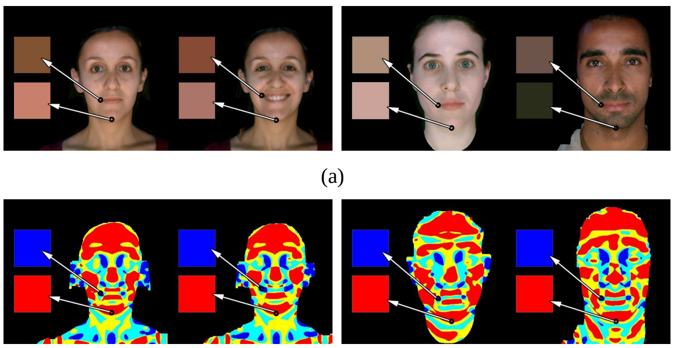
where d_A refers to the *appearance distance* and d_S refers to the *structural distance* of the patterns to be matched. The $\lambda \in [0, 1]$ parameter represents a weighting factor.

[2] In the general formulation, different mapping functions $f : V_1 \rightarrow V_2$ between the vertices of the graphs G_1 and G_2 are possible.

[3] In particular, $f : V_1 \rightarrow V_2$ is an *isomorphism* if $a, b \in V_1$ are adjacent whenever $f(a), f(b) \in V_2$ are.

1) Appearance Distance: As described in section II, for every geometry point (x, y, z) of the raw mesh, a (u, v) coordinate is assigned in the corresponding 2D image. Thus, every point has an $\mu_{UV}(v)$ attribute, related to the (R,G,B) color of the texture image. This attribute is part of vertex cost, since, when matching different expressions of the same person, it is expected that corresponding points have similar colors.

When matching faces from different people, though, this attribute alone is not sufficiently reliable (Figure 3(a)). Therefore, we also consider a $\mu_{HK}(v)$ curvature attribute, based on the *HK map* [17], that classifies surface regions by the computation of its mean (H) and Gaussian (K) curvature maps (Table I). As shown in Figure 3(b), corresponding points are usually part of the same HK classification.



(a)

(b)

Figure 3. Points comparison considering texture color (a) and HK map (b).

Table I
HK CLASSIFICATION.

	$K < 0$	$K > 0$
$H < 0$	• Hyperbolic convex region	• Elliptical convex region
$H > 0$	• Hyperbolic concave region	• Elliptical concave region

Briefly, the appearance cost d_A is formulated as follows:

$$d_A(v_m, v_i) = \lambda_A d_{A_{UV}}(v_m, v_i) + (1 - \lambda_A) d_{A_{HK}}(v_m, v_i) \quad (2)$$

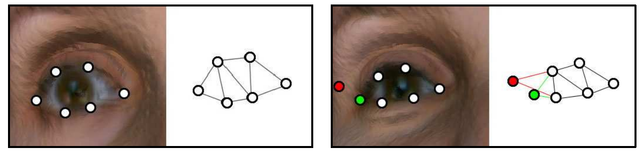
with

$$d_{A_{UV}}(v_m, v_i) = \frac{EuclideanDistance(\mu_{UV}(v_m), \mu_{UV}(v_i))}{C_A},$$

$$d_{A_{HK}}(v_m, v_i) = \frac{EuclideanDistance(\mu_{HK}(v_m), \mu_{HK}(v_i))}{C_A}.$$

where C_A is a normalization constant and $\lambda_A \in [0, 1]$ is a weighting factor[4].

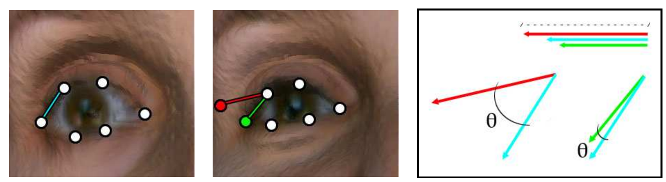
2) Structural Distance: When considering the attributes $\mu_{UV}(v_m)$ and $\mu_{HK}(v_m)$ on the cost computation, we are only measuring the points local similarity. It is also important, though, to measure the structural similarity of the patterns to be matched. In order to extract the structural information from the graphs, is not enough to minimize the vertices cost, but it is also necessary to minimize the edges cost, using an auxiliary structure called *deformation graph* (G_d). Initially, the deformation graph is simply a copy of the graph model. Then, for all candidate points of the input mesh, we evaluate how much the point insertion distorts the graph model, and select the points that minimize this distortion (Figure 4).



(a) Model points (b) Input candidates

Figure 4. After the selection of the model points (a), we evaluate two input candidates to match the eye corner (b). Visually, it is clear that the red point yields a greater distortion on the deformation graph than the green point.

In order to measure the distortion, we assign an $\nu(e)$ attribute to the graph edges, related to the vector that represents the edge e. The edge cost when matching points v_m and v_i is related to the mean similarity of each adjacent edge to v_m and each adjacent edge to v_i in the deformation graph. This similarity measure considers the size and angle between the vectors representing the edges to be matched (Figure 5).



(a) Model edge (b) Input candidates (c) Edge comparison considering dimension and angle.

Figure 5. The model edge (a) is compared with two candidate edges (b). Although the dimension of the edges is similar, the angle between the model edge and the red edge is greater that the one considering the green edge, yielding a higher cost (c).

The structural distance d_S is computed in the following way:

$$d_S(v_m, v_i) = \frac{1}{|E(v_d)|} \sum_{e_d \in E(v_d)} c_{vec}(\nu(e_m), \nu(e_d)) \quad (3)$$

where c_{vec} represents the vectorial cost, computed as follows:

$$c_{vec}(\vec{v_1}, \vec{v_2}) = \lambda_S d_{S_{ang}}(\vec{v_1}, \vec{v_2}) + (1 - \lambda_S) d_{S_{mod}}(\vec{v_1}, \vec{v_2}) \quad (4)$$

with:

$$d_{S_{ang}}(\vec{v_1}, \vec{v_2}) = \frac{|cos\theta - 1|}{2},$$

$$d_{S_{mod}}(\vec{v_1}, \vec{v_2}) = \frac{||\vec{v_1}| - |\vec{v_2}||}{C_S}.$$

where C_S is a normalization constant and $\lambda_S \in [0, 1]$ is a weighting factor.

3) Results: We applied two types of tests using the 210 meshes acquired by the scanner, with $\lambda_S = \lambda_A = 0.5$. At first, we took the neutral faces of the individuals as models (with manually selected points) and tried to match the corresponding landmarks with the other six expressions of

[4]The Euclidean distance between $\mu(v_1) = (R_1, G_1, B_1)$ and $\mu(v_2) = (R_2, G_2, B_2)$ is $\sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}$.

Figure 6. Some results of the structural matching. In all images, the face on the left represents the model (on which the points were manually selected) and the face on the right was automatically landmarked by the algorithm. Red points indicate mismatches from the ground truth. Although the results are shown in 2D, for simplicity, it is important to emphasize that it represents a 3D landmarking procedure.

the same individual. Then, we tried to match points between different individuals, using the same expression as model and input. Using as groundtruth a previous manual selection of the 48 points for all the individuals, we computed the algorithm errors for the two tests, reported in Tables II and III.

Table II
MEAN ERROR OBTAINED WHEN MATCHING THE NEUTRAL EXPRESSION AND THE SIX EXPRESSIONS OF THE SAME INDIVIDUAL.

Joy	Sadness	Surprise	Anger	Disgust	Fear
6%	0.6%	10%	3%	3%	6%

Table III
MEAN ERROR OBTAINED WHEN MATCHING THE SAME EXPRESSIONS BETWEEN DIFFERENT INDIVIDUALS.

Neutral	Joy	Sadness	Surprise	Anger	Disgust	Fear
3%	4%	5%	9%	8%	7%	9%

The method obtained very good results on landmarking the face contour, nose, eyes and eyebrows but had problems on the mouth region, when the differences between the model and input mouths were too sharp (Figure 6). Therefore, we used the algorithm in a semi-automatic way, using a simple interface to view the 3D matching result and remark incorrect points, which still represents a process that is considerably less exhausting than the original step of manually selecting the entire set of 48 points.

B. Subdivision Process

After obtaining meshes with a set of sparse corresponding points, using the approach previously described, we needed to compute the full dense correspondence between the models, in order to complete the registration step. We followed the approach of Guskov *et. al* [18] and Golovinskiy *et. al* [19] in which the initial model of landmarks is subdivided and re-projected, as described in Figure 7.

In the dataset, the process is performed using the same initial model with the same number of subdivision steps, i.e. the resulting meshes share the same topology, since every initial triangle is divided considering the same strategy.

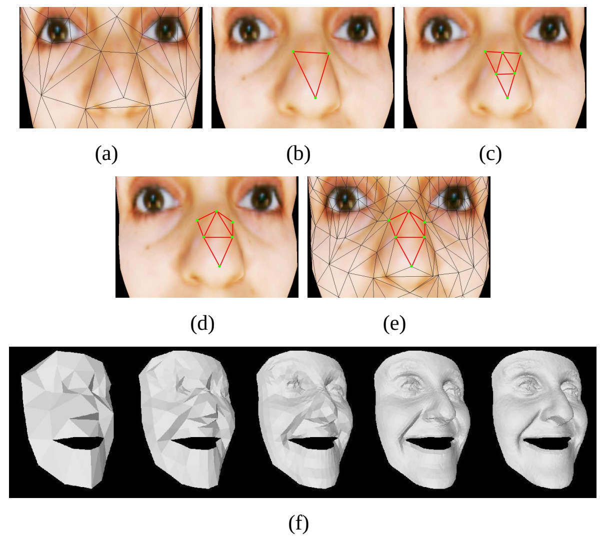
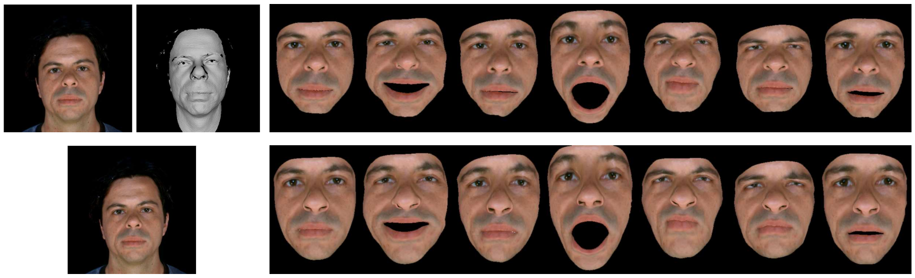


Figure 7. The subdivision process works as follows: a initial model is defined (a), a triangle is selected (b), new vertices are generated (c), the new vertices are projected on the raw mesh using its normal vectors (d), and the process is repeated for every triangle (e) refining the raw mesh at each step (f).

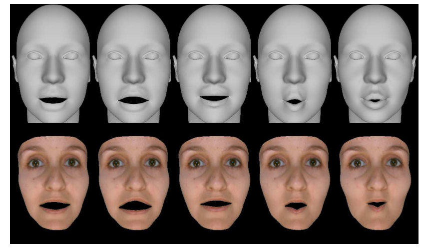
IV. ANIMATION BASED ON LINEAR INTERPOLATION

Once the models are subdivided, for every vertex that is part of the neutral model, it is possible to locate the corresponding vertex in all the expressions. Therefore, the variation of a given pose with respect to the neutral face is computed by the linear difference between every vertex of the pose and its corresponding vertex at the neutral face. After computing the shifts for all vertices, it is possible to build a matrix $M^k_{n\times 3}$ for each of the k poses containing the translations that should be applied to the vertices of the neutral face. The interpolation process gradually increments this variation to its final value and an animation is obtained, given the geometry of all the expressions of an individual I_A of the dataset (Figure 9).

The knowledge obtained in this process can be used to animate other people from the dataset given only the geometry of the neutral expression. If we subdivide a neutral face in the same way, its topology will be equivalent to I_A's topology.



(a)



/a/ /ɛ/ /i/ /ɔ/ /u/

(b)

Figure 8. Results of the expression transferring from the person in Figure 9 given texture and geometry information (a, top) and only texture (a, bottom) and visemes transferring from artificial models (b).

0%

50%

110%

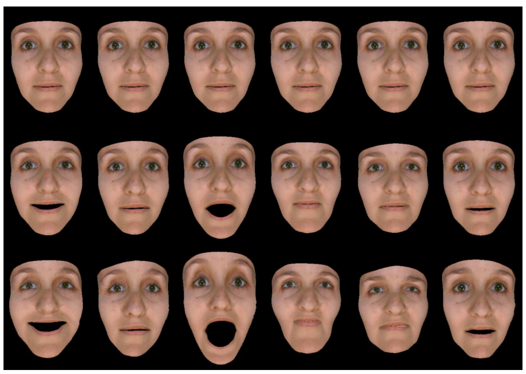


Figure 9. Interpolation of the six expressions for an individual.

Therefore, we can use the set of matrices M^k previously computed to transfer the animation from I_A to the target neutral face. Besides transferring expressions between people from the dataset, we can consider any model that contains the neutral geometry information to produce a similar animation. We are able to use and animate the result of reconstructed models [14] and to transfer new expressions from artificial models to the real models used. For the second application, we transfer visemes expressions in order to use the real models as part of a speech synthesis application. Expression transfer results are shown in Figure 8.

Facial animation is not the only application that requires the generation of a one-to-one mapping between facial meshes. Once we have a set of models with the same topology, we can also generate automatic caricatures. As in the interpolation method, where we increase the displacements between the neutral face and an expression, we are able to emphasize facial features by increasing the displacements between an individual face and an average face, generated by the mean geometry of all the dataset individuals [1] (Figure 10).

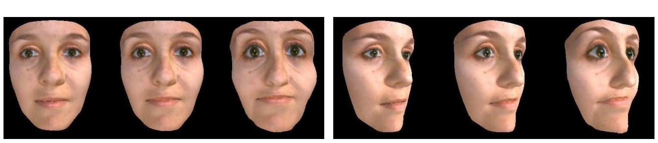


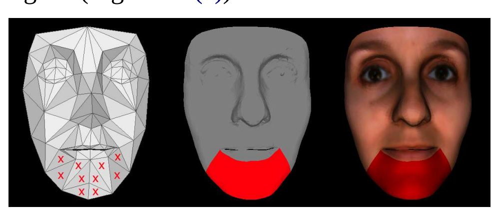
Figure 10. Automatic caricaturization by emphasizing facial regions.

V. Animation Based on Parametric Model

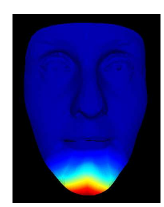
Although this work has started addressing the interpolation method, the landmarking and subdivision procedures could also be used to implement another method, based on the MPEG-4 animation standard. The method generates facial expressions and visemes given the neutral faces of the same real dataset. The MPEG-4 standard defines a set of feature points and animation parameters that acts on these points. The parameters modifies small portions of the face so that, when applied together, are able to generate the new poses. For example, in order to raise an eyebrow, we apply a parameter that moves the eyebrow center up in the y direction.

It is clear, however, that modifying only one point does not gives a good animation effect. It is necessary to define which neighborhood region around the point should be modified and the deformation function that should be applied in the region's points. Those issues are not addressed by the standard.

As in the previous method, the data is submitted to the same subdivision step. For this strategy, though, it is necessary to use another initial model, containing the MPEG-4 feature points[5]. We compute the neighborhood region using the initial triangulation of the subdivision process. By manually indicating a few triangles on the initial model, the subdivision step is able to determine, for each feature point, the whole corresponding region (Figure 11(a)).



(a)



(b)

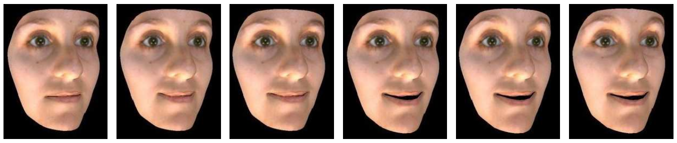
Figure 11. Computation of the jaw's region, using the initial triangulation followed by the subdivision step (a) and color map for its deformation (b).

Then, it is necessary to establish a deformation function that transfers the feature point deformation to its neighbors. In this aspect, it is desired that a large portion of the offset value is assigned to vertices that are close to the feature point, while

[5]The MPEG-4 defines 84 feature points, but some belongs to the inner mouth region and for practical purpose were discarded in this study.

distant vertices must suffer a small deformation. We computed the distances from the feature point and its neighbors and formulated a Gaussian based function. The application of this function is shown in Figure 11(b).

After the definition of the face deformation, an animation can be obtained by applying the appropriate parameters. Figure 12 shows a sequence of parameters applied in order to generate the joy expression and Figure 13 shows the method's result for all the facial expressions.



(a) (b) (c) (d) (e) (f)

Figure 12. Parametric generation of the joy expression from the neutral face (a). Parameters applied: raise corner lips (b), stretch corner lips (c), open mouth (d), raise eyebrows (e), lift cheeks (f).

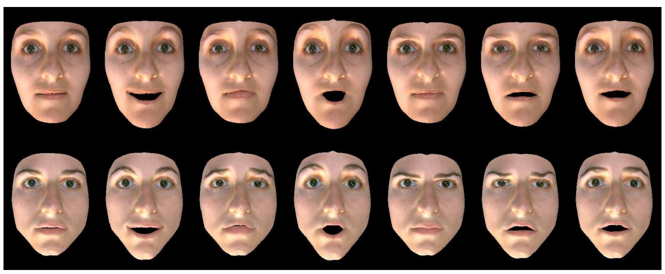


Figure 13. Results of the parametric expressions for two individuals.

The comparison between the animation quality of the two methods is subjective, since it depends on personal judgments. The fundamental difference between the two approaches is based on the fact that the interpolation model applies a global deformation to the mesh, while the parametric model generates small local deformations.

VI. Conclusion

Many studies require manual landmarking selection in order to register or normalize facial data. So far, there is not a representative solution that is able to compute a considerable number of 3D landmarks in a robust way.

We introduce a novel approach for facial landmarking, based on structural methods, and show how the sparse landmarks are used to compute dense correspondences, for facial data registration. Then, we propose interpolation and parametrization strategies to animate the registered facial meshes.

3D facial landmarking and registration are used for many applications, such as animation, reconstruction and facial recognition. Although focusing on the first, our work showed good results that allows it to be used to solve other problems in the previously mentioned areas.

Considerations: During our work, we published a conference paper [20] and a conference poster [21]. A new manuscript is being written for submission to a scientific journal.

Acknowledgment

The authors would like to thank FAPESP, CNPq, CAPES and FINEP for the financial support and professor Luiz Velho from IMPA-RJ for the data acquisition.

References

[1] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999.

[2] C. Basso, P. Paysan, and T. Vetter, "Registration of expressions data using a 3D morphable model," in *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*. IEEE Computer Society, 2006.

[3] V. Blanz, K. Scherbaum, and H.-P. Seidel, "Fitting a morphable model to 3D scans of faces," in *IEEE International Conference on Computer Vision*, vol. 2007. IEEE Computer Society, 2007.

[4] V. Blanz, C. Basso, T. Vetter, and T. Poggio, "Reanimating faces in images and video," in *Proceedings of EUROGRAPHICS*, 2003.

[5] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *AVSS '09: Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 296–301.

[6] J.-y. Noh and U. Neumann, "Expression cloning," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001.

[7] D. Vlasic, M. Brand, H. Pfister, and J. Popovic, "Face transfer with multilinear models," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*. ACM, 2006.

[8] B. Bickel, M. Lang, M. Botsch, M. A. Otaduy, and M. Gross, "Pose-space animation and transfer of facial details," in *Proc. of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, jul 2008, pp. 57–66.

[9] Y. Hu, M. Zhou, and Z. Wu, "A dense point-to-point alignment method for realistic 3D face morphing and animation," *Int. J. Comput. Games Technol.*, vol. 2009, pp. 1–9, 2009.

[10] T. Whitmarsh, R. C. Veltkamp, M. Spagnuolo, S. Marini, and F. T. Haar, "Landmark detection on 3D face scans by facial model registration," in *Proceedings of the 1st International Workshop on Shape and Semantics*, 2006, pp. 71–76.

[11] X. Lu and A. K. Jain, "Deformation modeling for robust 3D face matching," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 1377–1383.

[12] O. Çeliktutan, H. Çinar Akakin, and B. Sankur, "Multi-attribute robust facial feature localization." in *FG*. IEEE, 2008, pp. 1–6.

[13] H. Dibeklioglu, A. A. Salah, and L. Akarun, "3D facial landmarking under expression, pose, and occlusion variations," 2008, pp. 1–6.

[14] J. P. Mena-Chalco, "Reconstrução de faces 3D através de espaços de componentes principais," Ph.D. dissertation, IME-USP, December 2010.

[15] A. Noma, "Duas abordagens para casamento de padrões de pontos usando relações espaciais e casamento entre grafos," Ph.D. dissertation, IME-USP, 2010.

[16] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*.

[17] A. Colombo, C. Cusano, and R. Schettini, "3D face detection using curvature analysis," *Pattern Recogn.*, vol. 39, pp. 444–455, March 2006.

[18] I. Guskov, K. Vidimče, W. Sweldens, and P. Schröder, "Normal meshes," in *Proceedings of the Computer Graphics Conference 2000 (SIGGRAPH-00)*, S. Hoffmeyer, Ed. New York: ACMPress, Jul. 2000, pp. 95–102.

[19] A. Golovinskiy, W. Matusik, H. Pfister, S. Rusinkiewicz, and T. Funkhouser, "A statistical model for synthesis of detailed facial geometry," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 1025–1034, 2006.

[20] A. B. Mattos, J. P. Mena-Chalco, R. M. Cesar-Jr., and L. Velho, "3D linear facial animation based on real data," in *Sibgrapi 2010 (XXIII Brazilian Symposium on Computer Graphics and Image Processing)*, 2010.

[21] A. B. Mattos and R. M. Cesar-Jr., "3D facial animation based on structural registration," in *Workshops of Sibgrapi 2009 - Posters*, 2009.