AN AUTOMATIC APPROACH FOR FACIAL FEATURE POINTS EXTRACTION FROM 3D HEAD

Achraf Othman, Oussama El Ghoul and Mohamed Jemni
Research Lab. of Technologies of Information and Communication
Tunis College of Sciences and Technics in Tunis, University of Tunis
5 Av. Taha Hussein, BP56 Bab Mnara Tunis, Tunisia
{achraf.othman, oussama.elghoul}@utic.rnu.tn, mohamed.jemni@fst.rnu.tn

Keywords: Image processing, Computer vision, Segmentation, Edge and feature detection.

Abstract: This work presents a novel approach for 3D head segmentation as well as for extraction of facial features vertices. These tasks are the preliminary task for retrieval, recognition, classification and tracking processes. 3D head regions are detected from a cloud of points using the ratio of geodesic to Euclidean distances between pairs of vertices that combine neighbor coordinates and position information of the point. In this paper, we propose a new algorithm to extract automatically feature points inside the facial region based on the topology of 3D face. Furthermore, in order to validate our approach, we have run experimentations on several types of full 3D head that differ in number of points and surfaces, gender and ethnic. Also, to approve its robustness, we apply our approach on a benchmark of scanned faces containing 67 models.

1 INTRODUCTION

In recent years, a concurrent interest in facial animation has been increased. In fact, thanks to the technological growth in hardware development, especially that of graphics card, it becomes possible to synthesize facial expressions in short time. Each system developed its own characteristics and its own techniques of animation. But all these techniques were based essentially on the topology of the mesh which constituted the human head. Insufficient resources and computing speed were the main constraints (Shamir, 2008). In addition to the problems of processing, the deformations, applied on a model, were usually modelled manually which required skilled human resources. The reusability of key or meta-model was important and interesting, contrary to a system based on interpolation like Free-Form Deformation (FFD) (Sederberg & Parry, 1986). Our work is located in the first phase of the animation facial which consists to design a system able to extract automatically feature points from a 3D mesh based on the ratio of geodesic and Euclidean distances. This goal must be pass by a beforehand task called 3D head segmentation. A point concerning the design of virtual models and their mesh topologies will not be discussed here. In our implementations, we apply directly on the mesh generated automatically by free open source tools or 3D scanned face from benchmark (Moreno & Sanchez, 2004). The benchmark GavabDB contains 427 of 3D facial surface images corresponding to 61 individuals (45 male and 16 female), and there are 7 different images per each person. The whole set of individuals are Caucasian and most of them are aged between 18 to 40 years old. The paper is organized as follows: Section 2 is a review of previous works of facial feature vertices extraction from 3D head. The approach and the different steps of treatment are presented in the next section. Final conclusion is drawn in Section 3.

2 CONTRIBUTIONS

2.1 Problematic

The extraction of data from 3D mesh (matrix of vertices and surfaces) has quickly emerged as a preliminary step for post-processing. These extracted data are abstract and they indicate neither the rotation (orientation) nor the scale of face. Generally, the face has a common topology at the level of the ears, eyes and the nose. These areas are more detailed at the level of mesh size. In other words, these areas are dense in number of points and surfaces. These areas have an important number of vertices. So, we talk about density distribution of the mesh. The distances between vertices in these areas are minimal. Since we can consider the mesh as a graph related, where vertices are the summits and the Euclidean distance between a pair of vertices is the arc. The distance between two vertices, which are not direct neighbors, is the minimal distance passing through their neighbors. This distance is called Geodesic distance. This treatment can lead us to determine a mark of the face such as:

- The 2 vertices of ears are the X Cartesian axis.
- The middle of the two vertices of ears and the nose tip is the second Cartesian axis (Z axe).
- The perpendicular to precedent two axes is the third Cartesian axis (Y axe).

2.2 Ascertainment

Aiming to determine automatically feature vertices from the mesh of the face, we analyzed its topology. In a first step, we noticed that the distribution of vertices is not fair. The density of vertices should be distributed according to the surface curvature. Some face parts like nose, lips, eyes or ears comprises the largest density of vertices because it needs more geodesic details. The first solution we proposed, concerning the extraction of significant face parts, is to compute areas containing the largest number of vertices. Applying one of the clustering algorithms to determine these areas should be an interesting way. However, this technique shows a malfunction to the faces scanned or some processed faces, due to the regular distribution of the vertices densities in all the face curvature. The second ascertainment that we noticed is about the curvature of the mesh. In fact, the face has a particular geodesic surface in which the curve is more intense in the neighborhood of the eyes, less intense in the neighborhood of the nose and variable in the ears surfaces. The intensity of the curvature is defined as the bending of the curve or as the rate of the direction's change of its tangent vector. This characteristic encourages the use of a Watershed-based approach. The third ascertainment is based on an anatomical aspect; it confirms that the human face is symmetric. The symmetry does not allow to do the segmentation or to compute feature points; however it can be very helpful to refine the obtained results. Finally, we remark that there exists a very interesting property of the mesh that we can exploit to determine significant parts of the face. Indeed, the rate of the geodesic distance between two vertices divided by its Euclidian distance increase in the ears, the nose, the lips and the eyes surfaces (show Fig 2).

Based on these characteristics, we have built our approach, which is described in the next section.

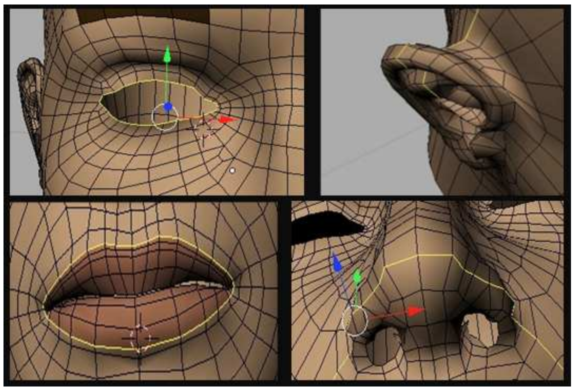

Figure 1: Parts of the face in which the geodesic distances are larger than the Euclidian distances.

2.3 Our approach

The approach that we propose is mainly based on the last property cited in the previous paragraph. We propose to compute the vertices that are close by Euclidean distance and far away by geodesic distance. After determining the significant zones of the mesh we propose to apply some geometric properties like the symmetry to compute feature vertices.

The input of the system is a mesh stored in two matrixes one contains the vertices list and second describes faces. The output is a vector containing the index of feature vertices. Previously, we described the problem of rotation and scale. To resolve this problem, we try to compute the high density distribution in the 3D head. The high density distribution contains vertices that are too close. To compute these distributions, we propose:

- Computing matrix of neighbors for each vertices.
- Computing matrix of Euclidean distances.
- Computing matrix of geodesic distances.
- Extracting key point for the three Cartesian axes.
- Extracting feature vertices.

To validate our approach, we conducted experimentations on GavabDB benchmark.

2.4 Neighbours' matrix

The matrix of neighbors is computed from matrixes of vertices and surfaces. It will be used for computing the shortest path between all pairs of vertices. We note that two vertices are neighbors if they are included in two surfaces at the same time.

2.5 Euclidean distance matrix

In digital geometry, with a three Cartesian axis, the Euclidean distance noteddist between two vertices V_i and V_j is computed as follow:

$$d = \sqrt{\left(P_{jx} - P_{ix}\right)^2 + \left(P_{jy} - P_{iy}\right)^2 + \left(P_{jz} - P_{iz}\right)^2} \qquad (1)$$

The Euclidean distance matrix noted M_{eucl} is computed between two vertices V_i and V_j. We iterate n times to browse all pairs of vertices. And for each vertex, we compute this Euclidean distance.

2.6 Geodesic distance matrix

The geodesic distance between two vertices is the shortest path from the source vertex to the target vertex through the neighbors. To compute the entire shortest path between all pair of vertices, we employ the algorithm Floyd-Warshall (Timothy, 2007) (Shalini & Markey, 2007). After computing neighbors, Euclidean distance and geodesic distance matrices, we try to locate the face, to determine the head pose and the scale. This step requires segmentation task. Segmentation algorithm is based on the ratio of Euclidean distance and geodesic distance. We iterate k times to extract vertices that compose clusters. Figure 6 shows the experimental result in a scanned face from GravabDB benchmark. In figure 2, we note that the edge is extracted as a cluster. This is due to the short distance between vertices in the edge of the face and to topology of the face. Our main goal is to extract feature points to animate a 3D head. So, we are interested only in full 3D head. We run our algorithms in scanned face just to approve robustness for noisy mesh in some cases. As a perspective, we may work to improve our technique.

2.7 Key vertices extraction

Key vertices play an important role in head pose orientation. Key vertices noted P_i are the vertices that compose the local three Cartesian axes in the mesh. These vertices are: $P_1$is the left ear, $P_2$is the right ear, P_3 is the midpoint between $P_1$and P_2 and $P_4$the nose tip. The four vertices will be extracted from the high density distribution in the mesh. In 3D head, the high density distributions are the areas of ears, eyes and frontal face (mouth and nose). Some properties for this distribution that two vertices are too close based on Euclidean distance and too far based on geodesic distance. So, if the ratio on Euclidean distance and geodesic distance for a set of vertices is near zero. For all pair of vertices we compute the ratio and we cluster the different distribution collected from the matrix of ratio. The results, in figure 2, show that the high density distributions are located in the ears and frontal face (composed by eyes, nose and mouth). Now, after extracting three density distributions from 3D head, we can extract $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ from the two clusters that are too far and equidistant to the rest clusters. $P_3(x_3, y_3, z_3)$ is the middle point. The vertex $P_4(x_4, y_4, z_4)$ is the equidistant vertex from P_1 and P_2. So we have: $\mathrm{dist}(P_1, P_4) = \mathrm{dist}(P_2, P_4)$. In some case, if we have not a symmetric mesh, we cannot find the coordinates of P_4. We add an epsilon error rate value in the formula. Experimental results on several scanned face and 3D head are the same for each model. Our algorithms are invariant to the pose, orientation, scale and translation. And this is approved mathematically by the ratio of the geodesic and Euclidean distance. Starting from this local 3D Cartesian axis, we compute head pose, scale and translation vector.

2.8 Feature vertices extraction

Extracting facial feature points such as eyes, mouth and nose plays an important role in many applications such facial animation, and this is our main goal. The proposed methods are based on the geometrical metrics of 3D mesh. At the beginning, we may extract the first feature point that allows us to compute the rest. The algorithm is based on the observation that feature points can be characterized by local as well as global conditions, in terms of their Euclidean and geodesic distances. We can extract now the nose tip. This vertex will be noted $P_{9.3}$ –same notation of MPEG-4 SNHC (Pandzic & Forchheimer, 2002) as follow:

1. Building a plane P passing fromP_3. The vector of P is noted $\overrightarrow{P_1P_2}$.
2. Projection of all the vertices near to P where $\mathrm{dist}(P, P_i) \leq \varepsilon$. ε is a small value aiming to reduce the error rate of computing.
3. Draw a line Δ through the too far points in P.
4. Compute the Euclidean distance from projected points to Δ plane.
5. The point $P_{9.3}$ is the far point to Δ and the nearest vertex to P_4.

To find the rest of vertices, we use the algorithm ‘peaks and troughs’ to locate the local extremum–

minima and maxima; in 3D mesh. Extrema are the largest value (maxima) or smallest value (minima), that a function takes in a point either within a given neighborhood. We apply the algorithm in 3D head from a plane P in the middle of the head (through the nose tip), the mid-plane between P and the left ear and the mid-plane between P and the right ear. The results are the feature vertices from 3D Head. Figure 8 shows the different steps from a source 3D head passing from the step of clustering, extracting the three local Cartesian axes and the final feature vertices. Experimental results were run on 13 full 3D head. The obtained results are very promising and that allow us to develop a full facial animation process in the future based on FACS (Ekman & Friesen, 1978) system or MPEG-4 (Dalong, 2002).

DISCUSSION AND CONCLUSION

In this paper, a novel and practical facial feature vertices system has been described, which combines several algorithms of feature detection with robustness to different mesh (gender, ethnic, scale, translation, pose head, number of vertices, number of surfaces). The proposed technique is based on 3D face/head segmentation computed from the ratio of Euclidean and geodesic distance. The procedure is fully automatic, while user modification is also allowed if necessary. The results can be used in a system for automatic facial animation. Experiment results testified the feasibility and validity of our method based on introducing geometrical metrics. There are several interesting directions in which our work can be extended, that address the main limitation of our work: reducing the execution time and robustness to the noisy model. The hidden goal of facial feature extraction is imitating human visual perception. Since this methods can neither be formalized nor measured mathematically. An empirical basis for research should be provided.

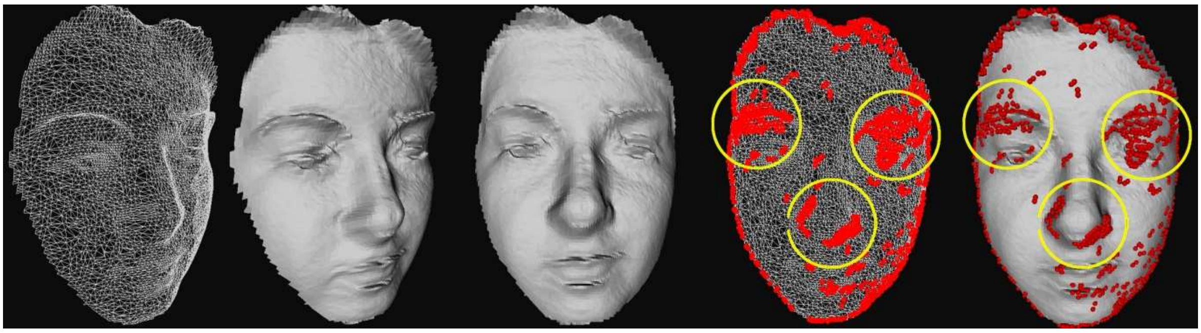

Figure 2: Extracted cluster in red applied to 3D scanned face. Yellow circles present the high density area in the face. Due to the noisy of the scanned mesh, boundaries of the face were considered also as a cluster (Vertices count=6292; Surface count=6152).

REFERENCES

Dalong, J., Zhiguo, L., Zhaoqi, W., & Wen, G. (2002). Animating 3D facial models with MPEG-4 FaceDefTables. *SS'02 Proceedings of the 35th Annual Simulation Symposium* , pp. 365 - 370.

Ekman, P., & Friesen, W. V. (1978). *Facial Action Coding System.* Palo Alto, CA: Consulting Psychologists Press.

Moreno, A., & Sanchez, A. (2004). GavabDB: A 3D Face Database. *Proc. 2nd COST Workshop on Biometrics on the Internet: Fundamentals, Advances and Applications* , pp. 77-82.

Pandzic, I. S., & Forchheimer, R. (2002). *MPEG-4 facial animation: the standard, implementation and applications.* J. Wiley.

Sederberg, T. W., & Parry, S. R. (1986, August). Free-Form Deformation of Solid Geometric Models. *SIGGRAPH'86* , pp. 151-160.

Shalini, G., Markey, M. K., Aggarwal, J., & Bovik, A. C. (2007, February). Three dimensional face recognition based on geodesic and euclidean distances. *Proceedings of the SPIE* .

Shamir, A. (2008). A Survey on Mesh Segmentation Techniques. *Computer Graphics Forum* , pp. 1539-1556.

Timothy, M. C. (2007, September). More Algorithms for All-Pairs Shortest Paths in Weighted Graphs. *Proceeding of STOC'07* .