

# SEMANTIC 3D FACE MESH SIMPLIFICATION FOR TRANSMISSION AND VISUALIZATION

Alessandro Colombo, Claudio Cusano, Raimondo Schettini

DISCo, Department of Computer Science, Systems and Communication.  
University of Milano Bicocca, Italy.

## ABSTRACT

Three-dimensional data generated from range scanners is usually composed of a huge amount of information. Simplification and compression techniques must be adopted in order to reduce transmission or processing time and to allow real-time visualization. In this paper we propose an approach to semantic simplification of triangular meshes representing faces. The algorithm is aimed to preserve facial features and can be used especially for face recognition systems purposes. In a first phase we detect salient regions using a 3D face detector based on curvature analysis and holistic classification. In a second phase, vertex decimation is applied to the mesh with different decimation parameters for salient and non salient-regions. We have tested our algorithm on a set of 150 acquisitions obtaining good visual quality meshes with approximately 90% or more of decimated vertexes.

## 1. INTRODUCTION

The interest in authentication and identification systems using biometrics has rapidly increased during the last few years. Face is one of the most interesting biometrics due to its non-intrusiveness property. Face is also one of the most challenging biometrics due to the high variability generated from lighting conditions, scale, pose, aging, obscurations, facial expressions and so on. In order to realize more robust face recognition algorithms, the use of 3D data has been investigated by many research groups. The incoming face recognition vendor test 2006 (FRVT2006) [1] and the face recognition grand challenge (FRGC) will also focus on evaluating performances using 3D facial scans. Apart from recognition algorithms issues, 3D face data brings other problems during the realization of a real commercial system. A face scan generated from a depth acquisition device is composed of a huge number of samples; nowadays devices can reach 300.000 or more samples per scan. Technical problems arise transmitting such an amount of data over the network due to bandwidth limit or real time constraints.

Moreover, it is possible that 3D face models have to be displayed on remote terminals with limited rendering capabilities. In this scenario, 3D data must be simplified in order to fit the system hardware constraints. Mesh simplification is the problem of reducing the number of mesh elements (i.e. triangles, edges and vertexes) minimizing the error between the original mesh and the simplified one. Several approaches have been proposed; the most popular are Shroeder's et al. vertex decimation [2], Hoppe's mesh optimization [3] and Garland's quadric-based simplification [4, 5]. For a comprehensive and detailed survey on simplification methods see for example [6].

Simplification algorithms are usually applied on the entire polygonal model given as input. Sometimes though, it could be better to apply different compression or simplification levels on distinct regions of the surface. This is exactly the case of 3D face models in face recognition systems. Although face images are cropped and segmented from the rest of the scene during the recognition pipeline, full-scene images are preferred for visualization. Either in case of processing or visualization, the interesting regions in an acquisition are those containing facial features like the eyes, the nose and the mouth. Visual attention experiments and saccades paths show that a human observer looking at a scene containing faces focuses most of his attention on facial features. For these reasons, simplification algorithms should be more conservative in those parts of the scene. Using these considerations, we have developed a semantic 3D face simplification algorithm. Our algorithm takes a 3D scene acquired by a depth scanner as input. In a first stage, the presence of faces is detected using a full 3D face detector [7] based on curvature analysis of the surface and a PCA classifier for face versus non-face discrimination. Once faces are detected, a bounding box approximating the volume of each face is passed to the simplification module.

Our simplification algorithm is a vertex decimator based on Shroeder's approach. Decimation thresholds for face regions are different from non-face regions in order to be more conservative in the former, but simplify the latter with a higher degree. The algorithm checks the position of each vertex against faces bounding boxes: if the vertex is contained in at least one bounding box then face decimation

parameters are applied; otherwise non-face parameters are considered.

## 2. SALIENT REGIONS DETECTION

Salient regions detection is achieved through a full 3D face detector described in detail in [7]; here we will give only a general overview. The face detector input is a three-dimensional scene taken from a depth scanner. Faces are assumed to look toward the acquisition device with some degree of freedom. In particular, faces can be freely oriented in the camera plane and limited rotations are allowed in the other rotation planes; i.e. the eyes and the nose must be visible from the camera. 3D data can be expressed either as a depth image or a polygonal mesh. Detection is accomplished using a two stage approach. In the first stage candidate faces are extracted through the curvature analysis of the scene. In the second stage, each candidate face is classified using the popular eigenfaces technique [8] applied to depth images.

### 2.1. Method description

The initial step of the detector is curvature computation. A new depth image is generated from an orthographic projection of the scene on the acquisition device image plane. Then, the gaussian (K) and mean (H) curvature maps are computed using the followings:

$$H(x_0, y_0) = \frac{(1 + f_y^2)f_{xx} - 2f_x f_y f_{xy} + (1 + f_x^2)f_{yy}}{2(1 + f_x^2 + f_y^2)^{\frac{3}{2}}},$$

$$K(x_0, y_0) = \frac{f_{xx}f_{yy} - f_{xy}^2}{(1 + f_x^2 + f_y^2)^2},$$

where  $f_x, f_y, f_{xy}, f_{xx}, f_{yy}$  are the first and second derivatives of the surface at image location  $(x_0, y_0)$ . The derivatives are computed using a biquadratic approximation of the local neighborhood of the point. By analyzing the signs of the mean and the gaussian curvature, we perform what is called an HK classification [9] of the points of the surface to obtain a concise description of the local behavior. HK classification classifies each point into eight different types: hyperbolic concave, hyperbolic symmetric, hyperbolic convex, planar, cylindrical concave, cylindrical convex, elliptical concave and elliptical convex.

In a face the eyes and the nose are among the regions with the highest curvature values and the most distinguishable one. These regions can be clearly defined by applying suitable thresholds. For each segmented region the average mean and gaussian curvature are then computed. More in detail, the system looks for the nose in the regions of positive mean curvature (convex regions); if the average mean curvature in the region exceeds the set threshold, the region is selected as a candidate nose. Candidate eyes, instead, are the elliptical concave regions for which the

average gaussian curvature is greater than a second threshold.

A set of candidate faces is built by grouping pairs of candidate eyes with the candidate noses. For each candidate face, the distances between the features are computed and compared with the corresponding distances in a typical human face. If the disagreement exceeds the set threshold then the candidate face is discarded.

In the second stage of the detector the portion of the surface below each candidate face is projected in a new range image in order to have the face turned upright, with both eyes on the same y coordinate, and with an angle of 45 degrees between the plane on which the three features lie and that of the equation  $z = 0$ . The nose is taken as the origin of the reference system. The range image is cropped with a mask to eliminate errors on face borders, so that all the images are equal in shape and size. Only the rigid portion of the face, including the eyes and nose, is considered.

These range images are then processed by a Principal Component Analysis classifier trained to discriminate between faces and non-faces; the approach is the same introduced by Turk and Pentland for face detection (see [8] for greater details). For each cropped range image an n-dimensional vector is generated by lexicographical construction. A set of initial training vectors is used to compute an eigenface basis that is a base of a space composed of lesser dimensions respect to the original image space. The eigenface basis has the axis oriented toward the directions of maximum covariance and preserves most of the information contained in the training set. When a new face vector is projected on the eigenface basis some information is lost. If a non-face vector is projected, the information lost is much higher because the basis is not trained to represent non-faces. The reconstruction error is used as a measure of "faceness" of an input pattern: high values of reconstruction error indicate non-face patterns; low reconstruction error values are associated to face patterns.

## 3. MESH SIMPLIFICATION

The mesh simplification algorithm is based on vertex decimation and is guided by the results of the face detection module. The input of the algorithm is composed of the initial triangular mesh of the scene, the location and orientation of the faces detected by the face detector module and the simplification parameters for face and non-face regions.

### 3.1. Vertex decimation

Vertex decimation has been proposed by Shroeder et al. [8] as a two step iterative approach for mesh simplification. The first step of the algorithm is vertex classification. Each vertex of the mesh is classified into five types: simple, complex, boundary, interior edge and corner. Simple

vertexes are those located in smooth regions. Complex vertexes are singular surface points, i.e. the surface is not differentiable in those points. Boundary points are located on mesh borders. Interior edge borders are located along a mesh sharp edge. The sharpness of a mesh edge is decided using a user-specified angle threshold. The angle is that formed by the normals of the two triangles along the edge. Finally, corner points are located on the intersection of two or more rough edges. Once the vertex classification is completed, only simple interior and boundary edges are considered as candidates for decimation.

The second phase of the algorithm is an iterative one. All candidate vertexes are analyzed using appropriate decimation criteria. If a vertex satisfies the criterion, then it is decimated. The algorithm stops when no more vertexes satisfy the decimation criteria.

Decimation criterion for simple vertexes is the distance to the plane. The plane approximating the neighbor vertexes is computed and the distance from the vertex to be decimated is considered. If the distance is higher than a given threshold the vertex is deleted.

Boundary and interior vertexes are decimated using the distance to edge criterion. The line defined by the two neighbor vertexes along the boundary or the sharp edge is considered and the distance to the vertex is computed. An additional uses-specified threshold determines if this distance is relatively small for preserving the vertex.

### 3.2. Triangulation

Unlike Shroeder et al. triangulation approach, we decided to adopt the vertex collapsing primitive for vertex elimination. The vertex collapsing primitive eliminates a vertex collapsing it on a neighbor one. When a vertex is collapsed some kind of artifacts can occur, namely sliver triangles, mesh inversion and topology changes. We have adopted different checks for each type of these artifacts.

Sliver triangles are very thin triangles that could cause problems due to numerical issues, so a good triangulation algorithm should avoid them. In our approach, when a vertex has to be collapsed, we compute, for each triangle of the new triangulation, a measure of how much equilateral it is. We have used the triangle compactness measure proposed by Guéziec [10]:

$$\gamma = \frac{4\sqrt{3}\omega}{l_1^2 + l_2^2 + l_3^2},$$

where  $l_1, l_2, l_3$  are the edges and  $\omega$  is the area. When  $\lambda$  equals 1 the triangle is perfectly equilater while a value of 0 represents a triangle with collinear vertexes. A user-specified threshold is used to detect triangles with low compactness. If a triangulation contains sliver triangles, another neighbor vertex is considered for collapsing the vertex to be decimated.

Mesh inversion occurs when one or more triangles of the triangulations inverts the direction of its normal so that the mesh folds over itself. We have used a simple check described in [4]. Let's call **a** the vertex to be decimated and **b** the vertex where **a** will be collapsed. For every triangle around **a**, excluding those in common with **b**, there is an edge opposite to it. For each edge we place a plane perpendicular to the triangle through the edge; the position of **b** must lie in the same side of the planes as **a**.

Topology changes can be detected using the link condition [11]. Roughly speaking, the link conditions checks the intersection between the neighborhood mesh elements set of vertex **a** and that of vertex **b** in order to prevent the generation of degenerative cases. For a more practical but equivalent check see [3].

### 3.3. Semantic face decimation

Semantic face decimation is achieved combining face detector results and the vertex decimation algorithm. The user must specify two sets of decimation parameters: salient regions parameters (S) and non-salient regions parameters (NS); see Table 1 for a summary.

Each vertex of the mesh is classified as salient or non-salient checking its containment in one of the detected faces bounding boxes. Once classified, vertexes are analyzed by the decimation algorithm as explained in section 3.1; the appropriate set of parameters is selected each time.

## 4. RESULTS AND CONCLUSIONS

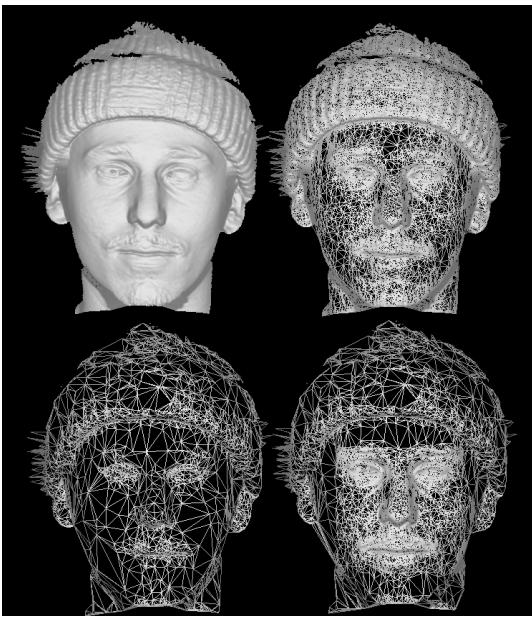
We have tested our algorithm on a set of 150 acquisitions captured by a laser depth scanner. All the acquisitions represent frontal faces from zero to two, while some acquisitions contain also some non-face elements like scarves, hats, little statues etc. Input meshes are composed of a number of vertexes ranging from 100.000 to 150.000.

In order to evaluate results quality we have chosen a human-based recognition scenario where the acquired 3D face models are sent to the user and then visualized for recognition. The user is able to rotate the face model so to explore it from any point of view. Results have been evaluated by a set of 5 persons.

For each original mesh we have generated three decimated versions: standard decimation with high quality parameters, standard decimation with low quality parameters and finally semantic decimation with low quality parameters for non-salient regions and high quality parameters for salient regions. Figure 1 and 2 show an example of a decimated face. Low quality parameters generate in some cases unacceptable visual quality in the face region, but achieve the maximum compression levels, i.e. a mean of 97% of decimated vertexes.



**Figure 1:** Top-left: Original Mesh, 123.000 vertexes. Top-right: high quality decimation, 13.302 vertexes (89% decimated). Bottom-left: low quality decimation, 2.157 vertexes (98% decimated). Bottom-right semantic decimation, 6.640 vertexes (94,6% decimated).



**Figure 2.** Wireframe versions of the models presented in Figure 1.

High quality parameters generate always acceptable results but low compression is obtained (90% of decimated vertexes). Semantic decimation generates meshes that are in most of cases acceptable for a human observer because interesting regions maintain most of the original information; compression level is usually better than high quality decimation (95%). In all cases the identities of subjects were always recognized.

In some acquisitions users pointed out that the transition between salient regions and non-salient regions is visible;

this happens especially if the difference between high and low quality compression parameters is high. We plan to avoid these artifacts using some kind of interpolation of the decimation parameters.

We have also tested the algorithm with an automatic face recognition system [12] based on PCA applied to the range images generated from a projection of the triangle meshes. We have obtained no significant changes in identification rates using original, high, low and semantic versions of the face meshes; in all cases we have obtained approximately 91% of successes.

**Table 1: Decimation parameters summary. For each parameter a salient and non salient value is considered.**

Parameter	Meaning
$\alpha$	Rough angle threshold
$d_p$	Distance to plane threshold
$d_e$	Distance to interior edge threshold
$d_b$	Distance to boundary threshold
$C$	Compactness threshold

## 6. REFERENCES

- [1] Face Recognition Vendor Test 2006. <http://frvt.org>
- [2] Schroeder, W. J., Zarge, J. A., and Ensen, W. E. Decimation of triangle meshes. Computer Graphics 26 (1992), 65–70.
- [3] Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle. W. Mesh optimization. ACM SIGGRAPH 1993, 19-26.
- [4] Garland, M., and Heckbert, P. Surface simplification using quadric error metrics. SIGGRAPH 97 (1997), 209–216.
- [5] Garland, M. Quadric-based polygonal surface simplification. Doctoral Thesis. 1999.
- [6] Heckbert P., and Garland M. Survey of Polygonal Surface Simplification Algorithms. In SIGGRAPH 97 Course Notes: Multiresolution Surface Modeling.
- [7] Colombo A., Cusano C., Schettini R. 3D face detection using curvature analysis. Pattern Recognition 39 (2006), 444-455.
- [8] Turk, M., and Pentland, A. Eigenfaces for recognition. Journal of Cognitive NeuroScience 3 (1991), 71–86.
- [9] Besl P.J., Jain R.C., Invariant surface characteristics for 3-d object recognition in range images. Comput. Vision, Graphics Image Process. 33 (1986) 33–80.
- [10] Guézeic A., Taubin G., Lazarus F., and Horn W. Converting sets of polygons to manifold surfaces by cutting and stitching. In IEEE Visualization 98 Conference Proceedings, pages 383-390,553. Oct. 1998.
- [11] T. K. Dey, H. Edelsbrunner, S. Guha and D. V. Nekhayev. Topology preserving edge contraction. Publ. Inst. Math. (Beograd) (N.S.) 66 (1999), 23-45.
- [12] Colombo A., Cusano C., Schettini R. A 3D Face Recognition System using Curvature-based Detection and Holistic Multimodal Classification. Proc. ISPA 2005, 179 – 184.