International Journal of Image and Graphics
Vol. 4, No. 1 (2004) 1–14

MESH COMPARISON USING ATTRIBUTE DEVIATION METRIC

MICHAËL ROY*, SEBTI FOUFOU† and FRÉDÉRIC TRUCHETET‡

Le2i, Université de Bourgogne
12 rue de la fonderie, 71200 Le Creusot, France
**michael.roy@iutlecreusot.u-bourgogne.fr*
†sebti.foufou@u-bourgogne.fr
‡frederic.truchetet@iutlecreusot.u-bourgogne.fr

Received 3 June 2002
Revised 16 December 2002

We propose a mesh comparison method using a new attribute deviation metric. The considered meshes contain geometrical and appearance attributes (material color, texture, temperature, etc.). The proposed deviation metric computes local differences between the attributes of two meshes. A mesh comparison assessment can be done easily and quickly using this metric. The techniques proposed are applicable in a number of ways, e.g. 3D matching and registration, and the example described in the paper is the simplification of a surface by iteratively reducing its complexity according to an error metric. The results are presented showing the success of the algorithm through comparisons with other measures and with three different simplification algorithms.

Keywords: Mesh; attribute; assessment; deviation; simplification; quality.

1. Introduction

Current computer graphic tools allow for the design and visualization of increasingly realistic and precise 3D models. These models are numerical representations of both the real and imaginary worlds. Acquisition and design techniques of 3D models (modeler, scanner, sensor, etc.) usually produce huge data sets containing geometrical and appearance attributes. Geometrical attributes describe the shape and dimensions of the object and they include data relative to a point set on the object surface. Appearance attributes describe object surface properties such as colors, texture coordinates, normal vectors, etc. High quality meshes usually contain a high number of vertices and faces that cause non interactive rendering or high storage space. In recent years results have been presented in order to reduce the mesh complexity either by merging/collapsing elements or by re-sampling vertices. Mesh simplification algorithms use different error criteria to measure the fitness of the approximated surfaces. Usually, these algorithms do not return the measures of the error introduced while simplifying the mesh. Therefore, a mesh comparison tool would be useful to characterize mesh simplification algorithms.

In this paper, we present a mesh comparison method based on a new attribute deviation metric. This assessment allows one to compute the local differences between the attributes of two meshes. The primary advantages of our method are:

- *Generality*: the method manages meshes containing geometric features as well as other surface attributes such as material colors, texture, temperature, radiation, etc. Moreover the measurements are independent of the viewpoint and the attribute type.
- *Locality*: assessments are done for given points on the mesh surface. Assessment resolution can be increased by a surface sampling method.
- *Applications*: the method is suitable for numerical models from real scenes and for synthetic models. This mesh comparison method can be used for many applications: mesh simplification, reverse engineering (comparison between a CAD model and a numerical model of the real object), mesh segmentation, mesh processing algorithm characterization, etc.

In Sec. 2, we review the related work on mesh difference metric. In Sec. 3, we present the attribute deviation metric used for mesh comparison. Algorithms used for the implementation of the proposed mesh comparison method are summarized in Sec. 4. Results on mesh simplification quality are presented in Sec. 5.

2. Review of Related Work

Complex meshes are expensive to store, transmit, and render. A lower level of detail can be obtained by simplifying the mesh (reducing the number of vertices and faces). In most cases, the simplified surface is therefore different from the original surface. Many simplification algorithms use their own error metric to guide the simplification process. Cignoni *et al.*[1] have presented an overview of the techniques used to evaluate the error introduced by the mesh simplification process.

2.1. *Simplification algorithms and error metrics*

Many algorithms use geometric measurements of distance or curvature. Schroeder *et al.*[2] use a vertex-to-plane distance as the decimation criterion. Reddy[3] employs a function based on curvature to guide the simplification process. Klein *et al.*[4] apply an error metric based on the Hausdorff distance. Ronfard *et al.*[5] use two energy functions: local tessellation error and local geometric error. Guéziec[6] utilizes a tolerance volume as an error bound measure. Rossignac[7] uses an error bound metric based on distances to supporting planes. Lindstrom *et al.*[8] use a volume metric.

These algorithms simplify the geometry and ignore the distortion caused to other surface attributes (colors, texture, normals, etc.). Figure 1 shows the results of an example of mesh simplification algorithms. Figure 1(a) shows the original

(a) Original mesh with attributes (18 050 faces)

(b) Simplified with attribute management (1000 faces)

(c) Simplified without attribute management (1000 faces)

Fig. 1. Mesh simplification example. The simplification algorithm used in (b) manages appearance attributes, and the algorithm used in (c) does not.

mesh. The algorithm used in Fig. 1(b) manages appearance attributes, while the algorithm used in Fig. 1(c) does not. We see clearly in the last figure that the mesh aspect is highly deteriorated.

Thus, more complete algorithms are needed to manage mesh attributes during the simplification process. Hoppe[9] uses three energy functions to preserve surface geometry, scalar attributes and discontinuity curves. Instead of using an error measure, Cohen *et al.*[10] propose a geometric construction, called the simplification envelopes, to minimize the surface deviation. Garland *et al.*[11,12] use a quadric error metric based on the computation of vertex-to-plane distances. Hoppe[13,14] has improved the last technique for meshes for appearance attributes.

Toubin *et al.*[15,16] have presented a method for analysis and simplification of numerical models from real scenes with several appearance attributes (temperature, luminance, etc.). The analysis of these models is done with the quincunx wavelet transform. This technique allows the extraction of geometric and appearance data which are considered as important information. Thus the original model is simplified in order to conserve these data. After what have been developed by Toubin, we have looked for an error metric to assess the conservation of the important information. Currently there is no tool to assess the simplification error introduced on appearance attributes. We have previously presented a simplification quality assessment for appearance attribute.[17,18]

2.2. *Geometric error*

Cignoni's *et al.* "Metro tool"[19] allows the measurement of mesh simplification algorithm quality as a geometric error between the original and the simplified meshes. This error is reported directly on the mesh, which allows the visualization of the local error. The software also returns numerical values such as the mean error. The quality measurement is based on the point-to-surface distance.

Given a point p and a surface S, the point-to-surface distance $e(p, S)$ is defined as:

$$e(p, S) = \min_{p' \in S} d(p, p')\,, \tag{1}$$

where $d(p, p')$ is the Euclidian distance between two points in $\mathbb{R}^3$. The Metro tool is commonly used. However it can only measure the geometric error and does not manage meshes with appearance attributes.

2.3. *Texture deviation*

Geometric data simplification introduces a modification of texture coordinates. Thus, the texture is applied differently on the simplified mesh than on the original one, which generates a modification of the texture aspect on the simplified surface. This alteration of aspect is called: *texture deviation*. Cohen *et al.*[20] proposed a measurement of this deviation to guide the simplification process.

Cohen defines an application $F_i(p) = (u, v)$ that allows the texture coordinates (u, v) to be associated to the point p on the surface S_i. This application allows one to travel from geometric space to parametric space. The inverse application $F_i^{-1}(u, v) = p$ that gives the point p on the surface S_i with the texture coordinates (u, v) is also defined. Given two meshes M_a and M_b, their respective surfaces S_a and S_b, and a point $p_i \subset S_a$, the texture deviation $T(p_i, S_b)$ between p_i and S_b is defined as:

$$T(p_i, S_b) = d(p_i, F_b^{-1}(F_a(p_i)))\,. \tag{2}$$

The texture deviation is the distance between a given point on S_a and the point on S_b with the same texture coordinates. The measurement of texture deviation is suitable to guide a simplification process, but results have shown that it is not suitable for assessing the simplification quality (see Sec. 5).

3. Mesh Comparison

Different shape matching methods have been proposed in the literature.[21–24] The most famous mesh comparison metric is the Hausdorff distance, which gives a global comparison between two meshes. This method is not suitable in our case because we desire to highlight mesh regions sensitive to the simplification process. Thus, we need to develop a local comparison assessment.

Attributes are considered as vectors in the Euclidian space defined on all points of the mesh. Therefore a point is represented as an array composed of n attribute vectors $(\mathbf{a_1}, \ldots, \mathbf{a_n})$ with $\mathbf{a_i}$ as an attribute vector. We define an application $\mathbf{f_i}(p) = \mathbf{a_i}$ such that the attribute vector $\mathbf{a_i}$ of attribute i is associated with the point p. The mesh comparison method proposed in this paper is based on the difference assessment between the mesh attributes.

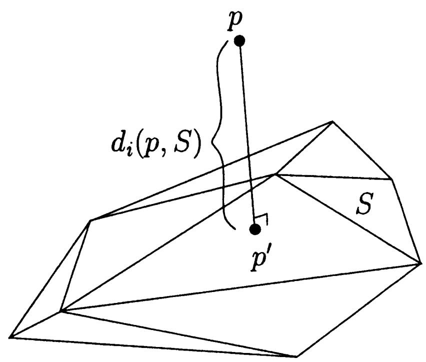


Fig. 2. Attribute deviation metric. The deviation $d_i(p, S)$ is the distance of the attributes i between a point p and a surface S. The point p' is the nearest point to p on the surface S.

3.1. *Attribute deviation metric*

Given a surface S and a point $p \in \mathbb{R}^3$, the deviation $d_i(p, S)$ between the attribute i of the point p and the surface S is defined as:

$$d_i(p, S) = \|f_i(p) - f_i(N_S(p))\| , \tag{3}$$

with $N_S(p) = p'$ the nearest point to p on the surface S. The attribute deviation $d_i(p, S)$ is the distance from the attribute i of the point p to the attribute i of the nearest point to p on the surface S. In the case of several points on the surface S having the same distance to the point p, the attribute deviation is the minimum distance between the attribute i of p to the attributes i of the nearest points to p on S. The attribute deviation metric scheme is presented in Fig. 2.

3.2. *Deviation assessment*

Given two meshes M_a and M_b, their respective surfaces S_a and S_b, and a set P of points $\{p_j | p_j \subset S_a \text{ and } j = 1, \ldots, m\}$, the deviation $D_i(M_a|_P, M_b)$ of the attribute i between $M_a|_P$ and M_b is defined as:

$$D_i(M_a|_P, M_b) = \{d_i(p_j, S_b) | j = 1, \ldots, m\} . \tag{4}$$

The deviation between two meshes uses the attribute deviation metric [Eq. (3)]. Mesh M_a is called the *reference mesh* and it is restricted to a point set taken on its surface. These points constitute the measurement points for the attribute deviation metric.

3.3. *Discussion*

Attribute deviation assessments allow one to highlight local differences between two meshes. Note that this assessment is guided by the geometrical correspondence between the meshes (the nearest point on a surface to a point on the other surface). Also note that the attribute deviation assessment is not symmetric. It is computed from a point set defined on the reference mesh surface. If the meshes are inverted, the results may be different. In practice, the average of both deviations $D_i(M_a|_{P_a}, M_b)$ and $D_i(M_b|_{P_b}, M_a)$ is considered. The results are often different but are relatively close.

As part of mesh simplification algorithm quality assessment, the geometric deviation assessment is the main measurement, since mesh simplification is essentially geometric. The deviation assessment of the other attributes can be used to quantify aspect modifications due to the simplification process. After an important simplification (90% or more reduced faces), we usually note the important local deviations of appearance attributes.

Note that Eq. (4) requires two meshes where one is sampled with a point set taken on its surface. Therefore we may take points only in the regions of interest. The measurement points may be vertices of the reference mesh or points taken on the mesh surface. The measurement resolution can be increased using a surface sampling algorithm (for example to compute deviation inside faces).

4. Algorithm Summary

Our mesh comparison method is based on the attribute deviation metric. For each point on the first surface, the nearest neighbors on the second surface are found, and deviations between them are measured. The performances of this method depend on algorithms implied in the different operations (nearest neighbor search, face sampling, etc.).

For a given point, the nearest point is computed efficiently by evaluating the point-to-surface distances. A regular grid of square cells is built, covering the bounding box of both meshes.[25,26] Each cell contains a list of all vertices included in the cell as well as all the faces intersecting the cell. This technique allows one to quickly find the nearest point on a surface for a given point. Note that the nearest point could be a vertex or a point on an edge or a point on a face.

We have developed a fast algorithm to sample a triangular face. This algorithm is based on the scan conversion algorithm.[27,28] The face sampling is performed in 3D space. In order to maintain the best accuracy, there is no 2D projection. This algorithm allows the generation of uniformly distributed points on a triangle in 3D space (see Fig. 3).

Given a face $f = (A, B, C)$, we define a local reference $(\vec{u}, \vec{v})$ as:

$$\vec{u} = \frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} \cdot \Delta \quad \vec{v} = \frac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|} \cdot \Delta\,, \tag{5}$$

where Δ is the sampling step. The reference $(\vec{u}, \vec{v})$ is defined on the plane formed by the face f. We generate the horizontal scan lines parallel to $\vec{u}$ and the vertical scan lines parallel to $\vec{v}$. The scan line numbers are defined as:

$$n_u = \frac{\|\overrightarrow{AB}\|}{\Delta}\,, \quad n_v = \frac{\|\overrightarrow{AC}\|}{\Delta}\,. \tag{6}$$

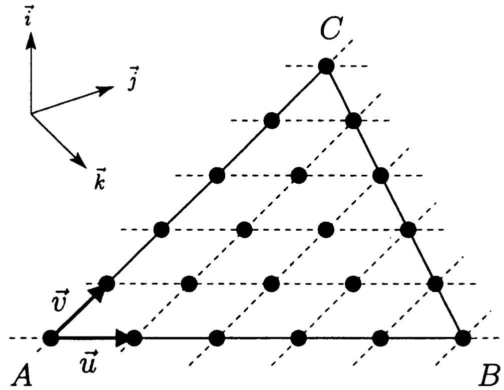


Fig. 3. Face sampling scheme.

The number of points per horizontal scan line can be determined by incremental computation using the slope of edge BC. With this sampling method, attributes can be easily computed for all samples using the Phong interpolation.[27] In order to obtain visual results of measured deviations on these points, a deviation image is created for every face. Deviation values are coded according to a color scale and images built for every face are packed in a standard, regular texture. We use the packing algorithm proposed by Cignoni[29] to build this texture.

5. Experimental Results

We have developed a mesh comparison software called MeshDev based on the attribute deviation metric presented above. This software is freely available online.[a] It requires two meshes as input, and returns numerical and visual comparison results. Numerical results contain characteristics of both meshes and statistics of the assessed deviation, and visual results allow one to highlight high deviation regions.

Table 1 shows an example of numerical results returned by the MeshDev software. The left box shows the mesh characteristics, the right box shows the statistics related to the measured deviation. More other statistical results can be given by MeshDev software.

Table 1. Example of numerical results returned by MeshDev software.

	Mesh M_a	Mesh M_b
Vertices	46,870	2,806
Faces	93,752	5,624
Area	22,090	22,124

	Deviation
Minimum	0.00013
Maximum	0.4933
Mean	0.0398
Variance	0.00085

[a] http://meshdev.sourceforge.net

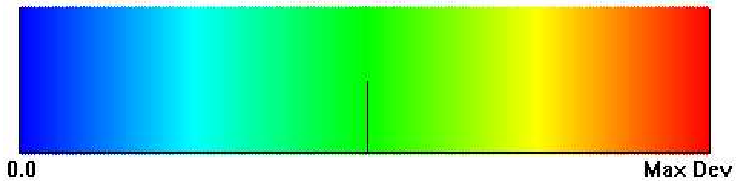
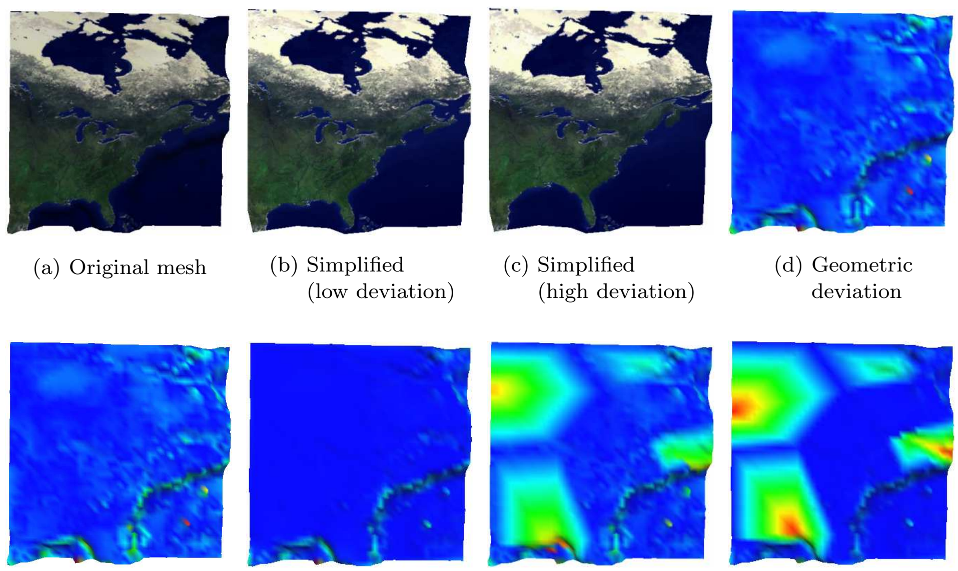


Fig. 4. Color scale used for visual representation of measured deviation.



(a) Original mesh (b) Simplified (low deviation) (c) Simplified (high deviation) (d) Geometric deviation

(e) Cohen's deviation for (b) (f) Attribute deviation for (b) (g) Cohen's deviation for (c) (h) Attribute deviation for (c)

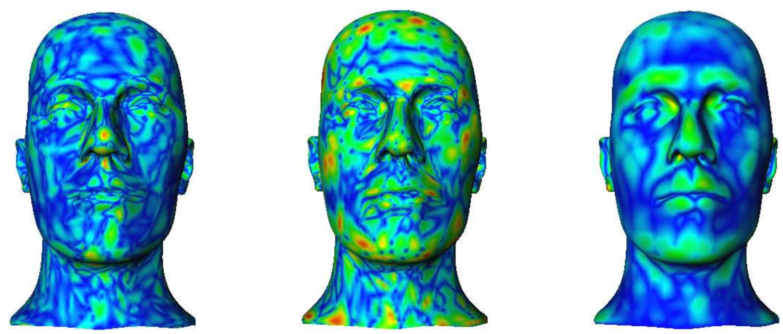
Fig. 5. Comparison between Cohen's measurement in (e) and (g) and the attribute deviation metric in (f) and (h) (the considered attribute is the texture coordinate).

Visual results are constructed by coloring the reference mesh according to the measured deviation. Figure 4 shows the color scale used: the left end is blue, representing minimum deviation, in the center is green which represents medium deviation, and red at the right end shows maximum deviation.

Figure 5 shows the visual results of geometric and attribute deviations computed using the MeshDev software. In this figure we compare the attribute deviation metric to Cohen's texture deviation measurement (see Sec. 2.3). Figure 5(a) shows the original mesh (3,972 faces). Figure 5(b) represents the simplified mesh (69 faces) with low texture distortion. Figure 5(c) represents the simplified mesh on which we have applied high texture distortion. Figure 5(d) shows the geometric deviation measured between the original and the simplified mesh. With low texture distortion, Cohen's measurement gives the same visual result as the geometric deviation (see Figs. 5(d) and 5(e)). Nevertheless, both measurements cannot be numerically compared. The attribute deviation metric returns only the effective texture deviation (see Figs. 5(e) and 5(f)). With high texture distortion, Cohen's measurement returns deviation values in regions where there is no distortion (see Fig. 5(g)). Attribute deviation metric gives more precise results than Cohen's measurement (see Fig. 5(h)). If there is no texture distortion, Cohen's measurement indicates a deviation coming from strictly geometrical distortion.

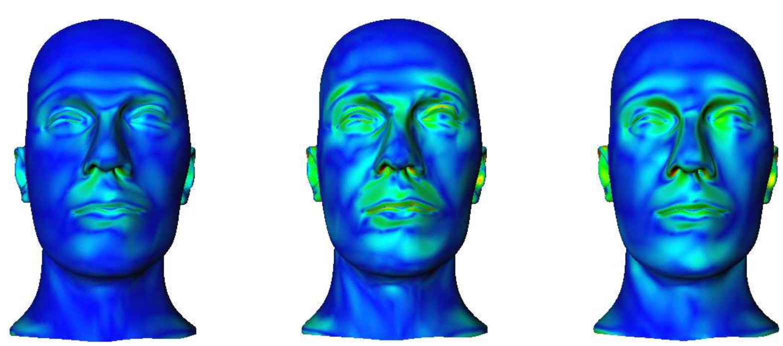
In our experiments, we used three simplification software programs:

- *QSlim*[b]: software developed by Michael Garland based on a quadratic error measurement.[11,12]
- *Jade*[c]: software developed by *Italian Visual Computing Group* based on a global error measurement.[30]
- *ProgMesh*[d]: software developed by *Paralelo* based on the progressive meshes from Hughes Hoppe.[9]



(a) QSlim (b) Jade (c) ProgMesh

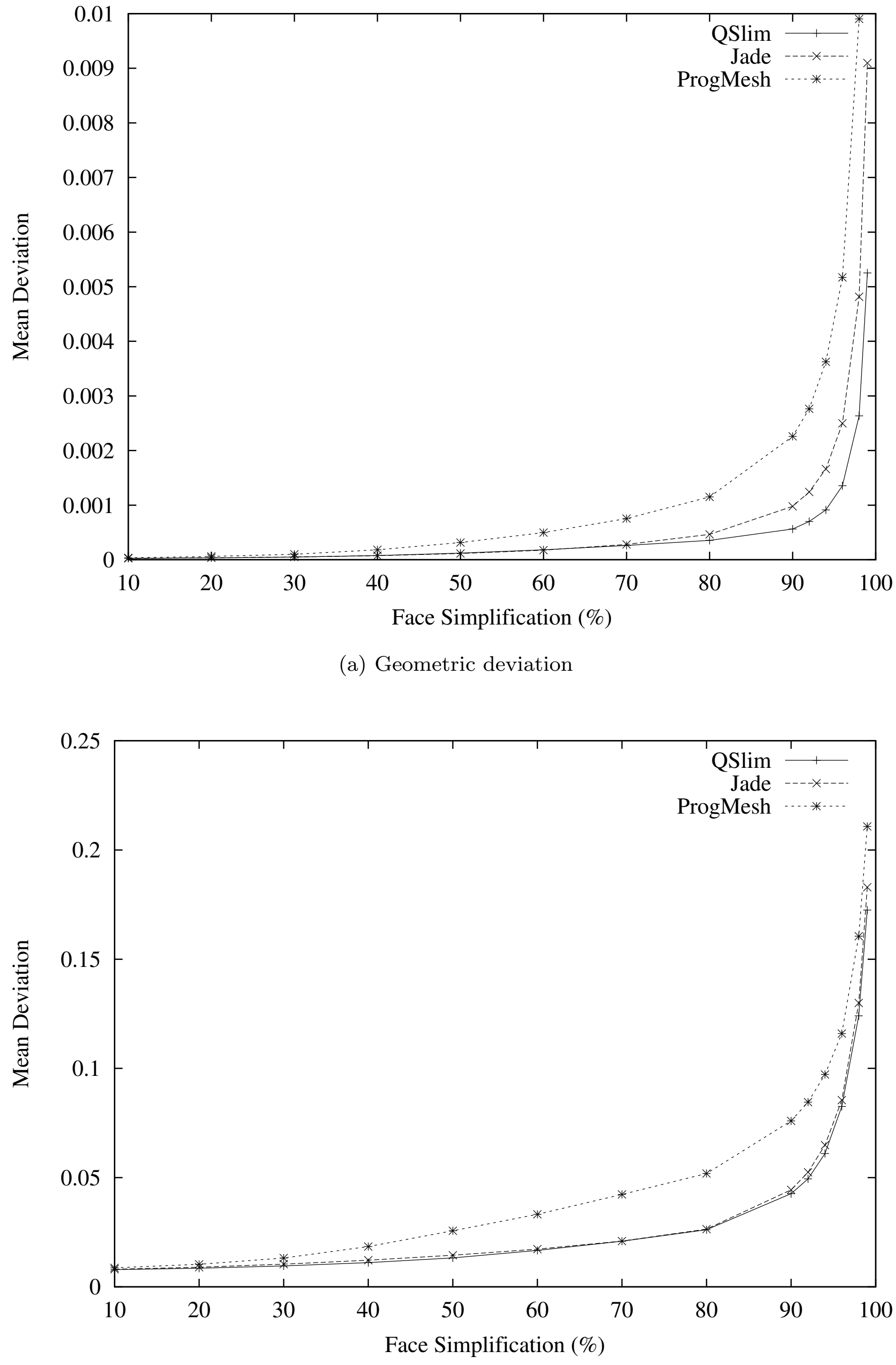
Fig. 6. Geometric deviation assessment for three different simplification algorithms.



(a) QSlim (b) Jade (c) ProgMesh

Fig. 7. Attribute deviation assessment for three different simplification algorithms (the considered attribute is the surface normal).

[b]http://graphics.cs.uiuc.edu/~garland
[c]http://vcg.iei.cnr.it/enhadecimation.html
[d]http://www.paralelo.com.br



(a) Geometric deviation

(b) Normal deviation

Fig. 8. Comparison between three mesh simplification algorithms.

Figures 6 and 7 show the visual results obtained with MeshDev software on a mesh simplified by these three simplification software programs. The deviation representation is normalized for each mesh, so the visual comparisons are not possible. We have chosen the normalized representation of deviations to highlight all measured values for each simplification algorithm. Figure 6 shows results of the geometric deviation and Fig. 7 shows the results of the attribute deviation in which the attribute considered is the surface normal (vertex normals are estimated by simply averaging face normals in the vertex neighborhood). *QSlim* generates a low deviation on the entire mesh but we find that there are some regions with high deviation (see Figs. 6(a) and 7(a)). *Jade* generates medium deviation on the entire mesh (see Figs. 6(b) and 7(b)). This algorithm simplifies the mesh by bounding the global error, while *ProgMesh* generates low and medium deviation on the entire mesh (see Figs. 6(c) and 7(c)).

Figure 8 represents the graphs of numerical results obtained from the simplified forms of the previous mesh. Simplified meshes are made using the three above cited software packages. Figure 8(a) shows the mean geometric deviation. Figure 8(b) shows the mean attribute deviation where the considered attribute is the normal. These graphs confirm the remarks made above. *QSlim* software obtained the lowest mean deviation in all cases. This software perfectly manages the appearance attributes during the simplification process and generates low deviation.

6. Conclusion and Future Work

We have presented the attribute deviation metric for approximating local differences between two meshes. This metric was used to build a mesh comparison program. The local measurement allows one to precisely view regions with high deviations. This is a great advantage compared to other methods that return global comparison. Geometric deviation is useful to assess shape differences. The deviation assessment of other attributes is efficient in assessing appearance modifications.

Since the attributes are considered as vectors in the Euclidian space, measurements in the real attribute space (such as RGB space for colors) would be more appropriate. Other metrics can be added to the attribute deviation metric (e.g. tesselation quality) in order to get a more general mesh comparison tool.

Acknowledgments

This work was made in collaboration with the IRIS Lab[e] (University of Tennessee). The authors acknowledge David Page,[f] Yiyong Sun,[g] Andreas Koschan, Mongi Abidi and Tadeusz Sliwa[h] for their help.

[e] http://imaging.utk.edu
[f] page@iristown.engr.utk.edu
[g] yiyong@iristown.engr.utk.edu
[h] t.sliwa@iutlecreusot.u-bourgogne.fr

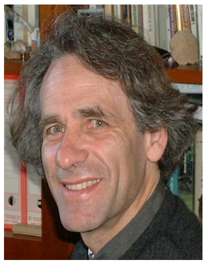
References

1. P. Cignoni, C. Rocchini, and R. Scopigno, "Comparison of mesh simplification algorithms," *Computer and Graphics* **22**(1), 37–54 (1998).
2. W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," in *Proceedings of SIGGRAPH*, pp. 65–70 (June 1992).
3. M. R. Scrooge, "Perceptually-driven polygon reduction," *Computer Graphics Forum* **15**(4), 191–203 (1996).
4. R. Klein, G. Liebich, and W. Straer, "Mesh reduction with error control," in *Proceedings of IEEE Visualization*, pp. 311–318 (October 1996).
5. R. Ronfard and J. Rossignac, "Full-range approximation of triangulated polyhedra," in *Proceedings of EUROGRAPHICS*, pp. 67–76 (1996).
6. A. Gueziec, "Surface simplification inside a tolerance volume," in *Second Annual International Symposium on Medical Robotics and Computer Aided Surgery*, pp. 132–139 (November 1995).
7. J. Rossignac, "Geometric simplification and compression," in *SIGGRAPH Course Notes* **25** (1997).
8. P. Lindstrom and G. Turk, "Fast and memory efficient polygonal simplification," in *Proceedings of IEEE Visualization*, pp. 279–286 (October 1998).
9. H. Hoppe, "Progressive meshes," in *Proceedings of SIGGRAPH*, pp. 99–108 (August 1996).
10. J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Books, and W. Wright, "Simplification envelopes," in *Proceedings of SIGGRAPH*, pp. 119–128 (August 1996).
11. M. Garland and P. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of SIGGRAPH*, pp. 209–216 (August 1997).
12. M. Garland and P. Heckbert, "Simplifying surfaces with color and texture using quadric error metrics," in *Proceedings of IEEE Visualization*, pp. 263–270 (1998).
13. H. Hoppe, "New quadric metric for simplifying meshes with appearance attributes," in *Proceedings of IEEE Visualization*, pp. 59–66 (October 1999).
14. H. Hoppe and S. Marchchner, "Efficient minimization of new quadric metric for simplifying meshes with appearance attributes," *Technical Report MSR-TR-2000-64, Microsoft Research* (June 2000).
15. M. Toubin, O. Laligant, A. Diou, F. Truchetet, E. Verrecchia, C. Dumont, and M. Abidi, "A multi-scale analysis for the characterization of 3D objects," in *Proceedings of SPIE Intelligent Robot and Computer Vision* **3522**, 414–423 (November 1998).
16. M. Toubin, D. Page, C. Dumont, F. Truchetet, and M. Abidi, "Multiresolution wavelet analysis for simplification and visualization of multi-textured meshes," in *Proceedings of SPIE Electronic Imaging* **3960**, 155–163 (January 2000).
17. M. Roy, F. Nicolier, S. Foufou, F. Truchetet, A. Koschan, and M. Abidi, "Assessment of mesh simplification algorithm quality," in *Proceedings of SPIE Electronic Imaging* **4661**, 128–137 (January 2002).
18. M. Roy, S. Foufou, and F. Truchetet, "Generic attribute deviation metric for assessing mesh simplification algorithm quality," in *Proceedings of IEEE International Conference on Image Processing*, pp. 817–820 (September 2002).
19. P. Cignoni, C. Rocchini, and R. S. Metro, "Measuring error on simplified surfaces," *Computer Graphics Forum* **17**(2), 167–174 (June 1998).
20. J. Cohen, M. Olano, and D. Manocha, "Appearance-preserving simplification," in *Proceedings of SIGGRAPH*, pp. 115–122 (August 1998).

21. R. Veltkamp and M. Hagedoorn, "State-of-the-art in shape matching," *Technical Report UU-CS-1999-27* (Utrecht University, 1999).
22. R. Veltkamp, "Shape matching: Similarity measures and algorithms," *Technical Report UU-CS-2001-03* (Utrecht University, 2001).
23. A. Johnson, "Spin-Images: A Representation for 3D Surface Matching," *PhD Thesis* (Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997).
24. A. Gueziec, "Meshsweeper: Dynamic point-to-polygonal-mesh distance and applications," *IEEE Visualization and Computer Graphics* **7**(1), 47–61 (2001).
25. W. R. Franklin and V. Akman, "Adaptative grid for polyhedral visibility in object space," *Computer Journal* **31**(1), 56–60 (February 1988).
26. W. R. Franklin, M. Kankanhalli, and C. Narayanaswami, "Geometric computing, uniform grid technique," *Computer Aided Design* **21**(7), 410–420 (1997).
27. J. Foley, A. Van Dam, S. Feiner, and J. Hughes, *Computer Graphics*: *Principle and Practice* (Addison-Wesley, second edition, 1990).
28. P. Heckbert, "Generic convex polygon scan conversion, clipping," in *Graphics Gems I* (Academic Press, 1990), pp. 84–86.
29. P. Cignoni, C. Montani, C. Rocchini, R. Scopigno, and M. Tarini, "Preserving attribute values on simplified meshes by re-sampling detail textures," *The Visual Computer* **15**(10), 519–539 (1999).
30. A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno, "Multiresolution decimation based on global error," *The Visual Computer*, *Springer International* **13**(5), 228–246 (1997).

Michael Roy is a PhD student at the Le2i, University of Bourgogne, France. He received a Master's degree in Electrical Engineering and Image Processing in Dijon, France in 2001. He's currently working in cooperation with the IRIS Lab, University of Tennessee, USA in the fields of 3D model scanning and processing. His technical interests include geometric modeling, multiresolution methods and computer vision.

Sebti Foufou is associate professor and member of the LE2i laboratory, University of Bourgogne, France. He obtained a PhD degree in 1997 from the University of Lyon 1, France, for a dissertation on parametric surfaces intersections. Since then is working on the field of CAD/CAM. His research interests include analysis of 3D meshes, surface modelling and the use of algebraic surfaces for CAD and computer graphics applications.



Dr. Fred Truchetet was born in Dijon, France, on October 13, 1951. He received the master degree in physics at Dijon University, France, in 1973 and a PhD in electronics at the same University in 1977. He was for two years with Thomson-CSF as a research engineer and he is currently full professor in Le2i, Université de Bourgogne (France) and CNRS, where he leads the image processing group. His research interests are focused on image processing for artificial vision inspection and particularly on wavelets transform, multiresolution edge detection and image compression. He has authored and co-authored more than 150 publications, three books and holds one patent. He is member GRETSI, ASTI, IEEE, SPIE, Chairman of spie's conference on wavelet applications in industrial processing and member of numerous technical committees of international conferences in the area of computer vision.