Engineering with Computers (2004) 20: 265–272
DOI 10.1007/s00366-004-0293-3

ORIGINAL ARTICLE

Rao V. Garimella · Mikhail J. Shashkov

Polygonal surface mesh optimization

Received: 28 July 2003 / Accepted: 12 April 2004 / Published online: 23 July 2004

Abstract A procedure has been developed to improve polygonal surface mesh quality while maintaining the essential characteristics of the discrete surface. The surface characteristics are preserved by repositioning mesh vertices so that they remain on the original discrete surface. The repositioning is performed in a series of triangular-facet-based local parametric spaces. The movement of the mesh vertices is driven by a nonlinear numerical optimization process. Two optimization approaches are described, one which improves the quality of elements as much as possible and the other which improves element quality but also keeps the new mesh as close as possible to the original mesh.

Keywords Polygonal surface mesh · Element quality · Jacobian condition number · Reference Jacobian matrices

1 Introduction

This paper describes a procedure to improve the quality of polygonal surface meshes by node repositioning while preserving the essential characteristics of the discrete surface and keeping the mesh close to the original configuration. The need for improvement of such meshes arises primarily in finite volume simulations where they form interior and exterior boundaries of general polyhedral meshes.

While previous research has focused on improving the quality of triangular and quadrilateral meshes [1–7], little attention has been paid to the improvement of polygonal meshes. Most of this work is devoted to smoothing (denoising) of a discrete surface represented by polygons (e.g., [8, 9]) rather than improving the quality of the polygonal elements in the surface mesh. In earlier work [7, 10], the authors presented a method for improving the quality of triangular and quadrilateral surface meshes in the absence of an underlying smooth surface. This paper extends and improves this technique to allow smoothing of surface meshes with general polygonal elements.

The rest of the paper is organized as follows. The next section describes the minimization of an objective function with respect to local parametric coordinates. That section discusses the element-based local parameterization, line search in local parametric coordinates, and moving vertices from one parametric space to another. The section titled “Optimization of surface mesh quality” discusses specific objective functions for optimizing the quality of surface meshes. Finally, the “Results” section presents several examples of optimization of surface meshes to demonstrate the capabilities of the methods.

2 Optimization with respect to parametric coordinates

Consider an objective function, $\Psi(\mathbf{x})$, defined in terms of the real coordinates, $\mathbf{x}$, of all the vertices of a surface mesh. The objective function is defined such that its minimization drives the mesh vertices to locations that improve the mesh, with respect to some quality measure. If this objective function is minimized directly, with respect to the real coordinates of the vertices, the search direction for the minimization may indicate vertex movement off the original surface mesh. To constrain the movement of the vertices to the discrete surface, the optimization must be performed with respect to the coordinates of the vertices in a 2D parametric space corresponding to the surface mesh. Assuming that there is no smooth surface underlying the discrete surface, one of several methods can be used to derive such a global parametric space from the surface mesh [11–15]. However, most of these methods involve substantial cost

R. V. Garimella (✉) · M. J. Shashkov
MS B284, Los Alamos National Laboratory,
Los Alamos, New Mexico 87545, USA
E-mail: rao@lanl.gov
E-mail: shashkov@lanl.gov

since they require the solution of a large system of equations that may be nonlinear.

In this work, instead of using a global parametric space derived from the entire mesh, nodes are repositioned in a series of local parametric spaces. The local parametric spaces are derived from a mapping of mesh edges and triangular facets of mesh faces to canonical elements in 2D space, as is commonly done in finite element methods [16, 17]. Vertices on the boundary of the surface mesh (i.e., on a model edge) move in parametric spaces of boundary edges of the original mesh. The parametric space of each boundary mesh edge is derived by mapping it to a unit line segment along the x-axis, giving rise to parametric coordinate $0 \leq s_0 \leq 1$. Vertices in the interior of the surface mesh (i.e., on a model face) move in parametric spaces derived from faces of the original mesh. The parametric space for a mesh triangle is derived using a barycentric mapping [17], resulting in parametric coordinates $0 \leq (s_1, s_2) \leq 1$ (Fig. 1a). Quadrilaterals and more general polygons are considered to be made up of triangular facets (Fig. 1b) and a parametric space is derived for each facet as before. The facetization of polygons is computed by choosing a central point and connecting it to the polygon edges. To choose the central point, a full quadric:

$$Z' = aX'^2 + bX'Y' + cY'^2 + dX' + eY' + f \qquad (1)$$

is first fitted to the polygon's vertices in a rotated frame $\{X', Y', Z'\}$ [18, 19] anchored at the centroid of the polygon. The central point to be connected to the polygon's edges is chosen as the point $(0, 0, f)$ in the rotated frame. If the polygon does not have enough points to fit a full quadric, additional points from the polygon's neighborhood are used.

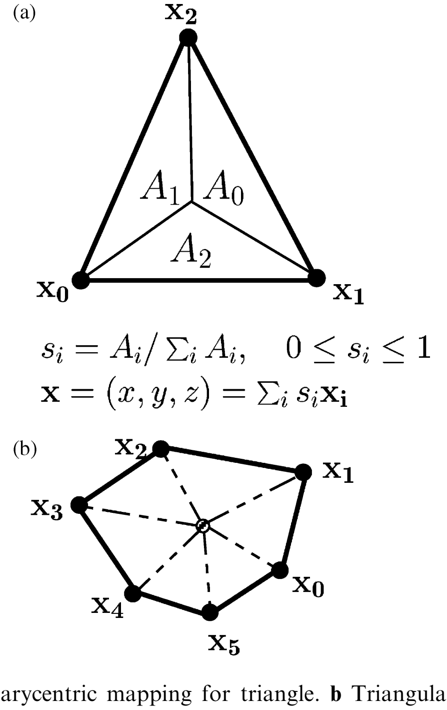
(a)
x_2
A_1
A_0
A_2
x_0
x_1
$s_i = A_i / \sum_i A_i, \quad 0 \leq s_i \leq 1$
$\mathbf{x} = (x, y, z) = \sum_i s_i \mathbf{x_i}$
(b)
x_2
x_1
x_3
x_0
x_4
x_5

Fig. 1 **a** Barycentric mapping for triangle. **b** Triangular facetization of polygon

The optimization procedure keeps track of the facet of the original mesh face that each vertex is moving in. The triangular facet in which a vertex is moving is referred to as the *base* triangle. The procedure also keeps track of the coordinates of the vertex in the parametric space of the base triangle. All objective function evaluations are done after mapping the parametric coordinates of the vertex in the base triangle to real coordinates. Also, the line search in the optimization procedure is conducted in the parametric space of the base triangle. The line search is used to find a step size, α, along a search direction, d, in the local parametric space, while respecting parametric bounds and mesh validity constraints. If an element becomes invalid during a line search, then the step size is scaled back and the optimization is restarted along a new search direction. If the line search takes the point out of the parametric bounds of the base triangle, the optimization is stopped, the adjacent triangular facet is found, and the optimization is restarted in the parametric space of the new base triangle. Additional details of the optimization procedure are given in [7].

3 Optimization of surface mesh quality

3.1 Condition number shape measure for polygonal mesh faces

The quality measure used for evaluating the shape of polygonal mesh faces is based on the condition number shape measure [20]. This measure is derived from the Jacobian matrix of an element mapping as described below.

Consider a vertex, V_i, connected to a set of edges, $\mathscr{E}(V_i)$, and faces, $\mathscr{F}(V_i)$, as shown in Fig. 2. Assume that one of the faces, $F_j \in \mathscr{F}(V_i)$, has edges $E_p \in \mathscr{E}(V_i)$ and $E_q \in \mathscr{E}(V_i)$ connected to vertex V_i. The triangle formed by edges E_p and E_q can always be mapped to a right-angled triangle in 2D space with V_i mapped to the origin, a unit vector, $\mathbf{e}'_p$, representing E_p along the x-axis and a unit vector, $\mathbf{e}'_q$, representing E_q along the y-axis in 2D space. Then, the *Jacobian matrix*, $\mathbf{J}_{ji}$, of the mapping of the triangle to the right-angled triangle in 2D space,

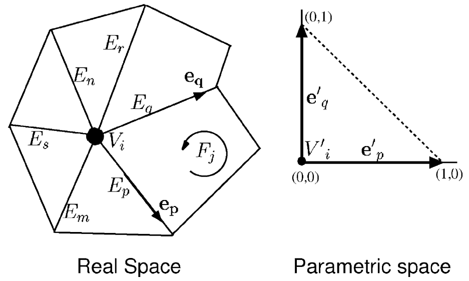

Fig. 2 Definition of edge vectors, $\mathbf{e}_p$ and $\mathbf{e}_q$, for calculating the Jacobian of an element F_j at vertex V_i

evaluated at vertex V_i, is given by $\mathbf{J}_{ji} = [e_p \quad e_q]$ where, $\mathbf{e}_p$ and $\mathbf{e}_q$ are 3D edge vectors, of lengths l_p and l_q, representing edges E_p and E_q, respectively. The condition number of the Jacobian matrix is defined as $\kappa(\mathbf{J}_{ji}) = \left|\mathbf{J}_{ji}^{-1}\right|_F \left|\mathbf{J}_{ji}\right|_F$, where $|\cdot|_F$ is the Frobenius norm of its matrix operand.

Since $\mathbf{J}_{ji}$ is a 3×2 matrix for a triangle in 3D, its inverse does not exist in the usual sense and a pseudo-inverse has to be calculated by singular value decomposition methods. On the other hand, the Jacobian matrix of a triangle in 2D space is a 2×2 matrix whose condition number can be calculated more easily as:

$$\kappa(\mathbf{J}_{ji}) = \frac{\left(l_p^2 + l_q^2\right)}{2A_j} \tag{2}$$

where A_j is the area of the triangle formed by E_p and E_q [20, 21]. This condition number is only a function of the triangle side lengths[1]; therefore, it is invariant with rotation of the triangle in the plane. Since there always exists a coordinate system in which an arbitrarily oriented triangle lies on one of its coordinate planes, it suggests that the condition number is also useful for measuring the quality of arbitrarily oriented triangles in space.

The condition number shape measure, as described above, measures the deviation of an element corner from a right-angled corner formed by unit edge vectors. Based on this property, a quality measure for a polygonal element may be defined as the sum of the Jacobian condition numbers at the polygon's corners. This sum reaches a minimum when the polygon is regular.

Note that that Jacobian condition number at an element corner is singular when the area, A_j, of the triangle formed by the corner is zero and is negative if the area is negative. Therefore, the Jacobian condition number is not a valid shape measure for polygons with concave or reentrant corners.

3.2 Condition number based optimization

Consider the minimization of a function defined as the sum of condition numbers of the face corners incident at a given vertex, V_i, as given below:

$$\begin{aligned}\psi_i^c(x_i) &= \sum_j \kappa(\mathbf{J}_{ji}(x_i)) \\ &= \sum_j \frac{l_p^2(x_i) + l_q^2(x_i)}{A_j(x_i)}, \quad j \in \{j|F_j \in \mathscr{F}(V_i)\}\end{aligned} \tag{3}$$

where l_p and l_q are the lengths of the respective edges, E_p and E_q, of face F_j connected to vertex V_i, and $\mathbf{x}_i$ is the coordinate vector of V_i.

The minimization of ψ_i^cattempts to smooth the distribution of face angles and edge lengths around a vertex since all the edge vector pairs are trying to reach equal length and form a right angle. Based on this property, a strategy can be formed for improving the quality of a mesh by minimizing a global condition number based objective function, Ψ^{c}, defined as:

$$\psi^c = \sum_j \psi_i^c, \ i \in \{i|V_i \in \mathscr{V}\} \tag{4}$$

where $\mathscr{V}$ is the set of all mesh vertices.

For efficiency reasons, the global function Ψ^{c} is, in reality, minimized by minimizing a local function, $\tilde{\psi}_i^c$, at each vertex. The value of $\tilde{\psi}_i^c$ at a vertex V_i is composed of all terms of Ψ^c that involve the coordinates of V_i. Therefore, $\tilde{\psi}_i^c$ is formed by visiting each element, F_j, connected to vertex V_i, and adding the Jacobian condition number of the element at V_i and the Jacobian condition numbers at both of its edge-connected neighbors in that element (see Fig. 3). Mathematically, this is written as:

$$\begin{aligned}\tilde{\psi}_i^c &= \sum_j \sum_k \kappa(\mathbf{J}_{jk}), \ j \in \{j|F_j \in \mathscr{F}(V_i)\}, \\ & k \in \{k|V_k \in \mathscr{V}(F_j) \cap \mathscr{V}(\mathscr{E}(V_i))\}\end{aligned} \tag{5}$$

Note that the presence of A_j in the denominator acts as a barrier that discourages vertex movements that make the triangle formed by E_p and E_q degenerate. However, for some optimization techniques, it may still be necessary to explicitly check if the optimization process is forcing the vertices to jump across the degeneracy barrier.

3.3 Reference Jacobian based optimization

3.3.1 Motivation

The global condition number minimization procedure allows mesh vertices to move along the surface as much as necessary to minimize the objective function, Ψ^c. However, in certain situations, it is of interest to keep the vertices of the original mesh as close as possible to their original locations while improving the shape of the mesh elements. Keeping the vertices close to their original positions facilitates accurate interpolation of

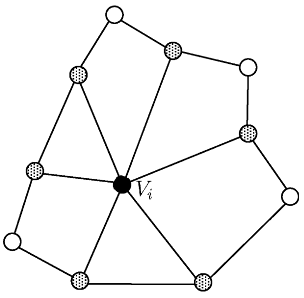

Fig. 3 Vertices involved in the local objective function expression, $\tilde{\psi}_i^c$, for V_i. The *shaded circles* along with the *black circle* (V_i) represent the vertices at which the Jacobian is computed for use in $\tilde{\psi}_i^c$. The *white circles* represent vertices whose real locations do not contribute to the Jacobian at V_i

[1] A_j is a function of the lengths of the triangle sides

solution data from one mesh to another, and also preserves mesh characteristics, such as refinement and anisotropy. The *reference Jacobian matrix (RJM) based optimization* [7, 10, 21, 22] is used here to achieve the multiple objectives of improving mesh quality and minimizing the change to the original mesh.

The RJM mesh improvement is a two-stage procedure, consisting of a series of local condition number based optimizations and a global RJM optimization, as described next.

3.3.2 Local condition number based optimization (step I)

This is the first stage of the RJM optimization strategy. In this step, the locally optimal position of each mesh vertex is computed with respect to the fixed position of its neighbors. The objective function for optimization is the local condition number function, $\tilde{\psi}_i^c$, described in Eq. 5, in the section titled "Condition number based optimization". However, in this step, the vertex is not moved to its locally optimal position. Rather, the optimal position of each vertex, described by a base face and the parametric coordinates of the vertex in the base face, is stored as a virtual position for use in the second stage of the mesh improvement procedure.

3.3.3 Reference positions, reference edges and the reference Jacobian matrix

The locally optimal position computed and stored for each vertex in the first stage of the procedure is known as the *reference position* for the vertex. After reference positions are calculated for all mesh vertices, two *reference edge vectors* are calculated for each edge in the mesh; each reference edge vector goes from the reference position of one vertex of the edge to the original position of the other. The idea of reference edges is illustrated in Fig. 4, where E_m is an edge with vertices V_a and V_b. The reference positions of V_a and V_b are $V_a{}^{\mathrm{R}}$ and $V_b{}^{\mathrm{R}}$, respectively. The two reference edge vectors for E_m are $(\mathbf{e}_m{}^{\mathrm{R}})_a$ and $(\mathbf{e}_m{}^{\mathrm{R}})_b$, where the outer subscript indicates which of the vertices is at its reference position.

Using the concept of reference edge vectors, it is now possible to define *reference Jacobian matrices* (RJMs) just as Jacobian matrices were defined for a mesh without reference positions. Therefore, if the edges of face F_j connected to vertex V_i are E_p and E_q, their reference edges are $E_p{}^{\mathrm{R}}$ and $E_q{}^{\mathrm{R}}$, and their reference edge vectors are $(\mathbf{e}_p{}^{\mathrm{R}})_i$ and $(\mathbf{e}_q{}^{\mathrm{R}})_i$ respectively, then the reference Jacobian of F_j at V_i is defined as $\mathbf{J}^{\mathrm{R}}_{ji} = \left[\left(\mathbf{e}^{\mathrm{R}}_p\right)_i \;\; \left(\mathbf{e}^{\mathrm{R}}_q\right)_i\right]$.

3.3.4 Global optimization based on reference Jacobian matrix (step II)

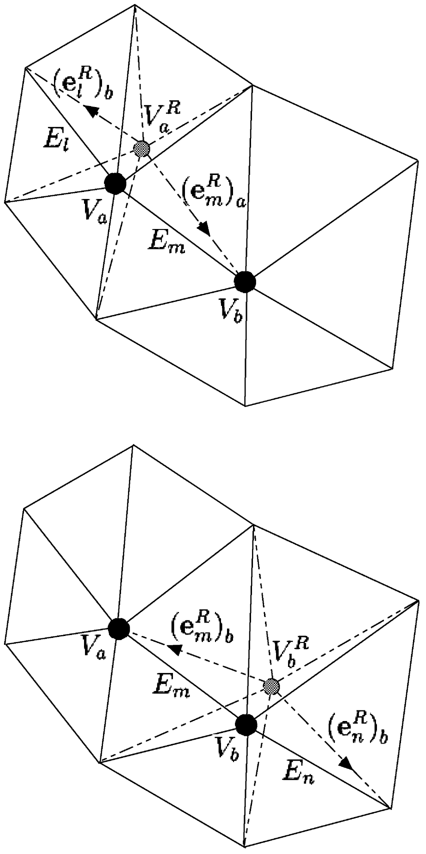

Fig. 4 Reference positions and reference edge vectors

The second stage of the mesh improvement procedure is a global optimization based on the definition of RJMs. The goal of this step is to find a valid mesh configuration such that each edge is in a compromise configuration between its pair of reference edges. It is expected that such a configuration for the edges will improve mesh quality since the reference edge vectors were formed by locally improving mesh quality at each mesh vertex. It is also expected that the optimized mesh will not deviate drastically from the base mesh, since each reference edge vector has one of its vertices at its original position and the other at the locally optimal position.

The objective function for the global optimization quantifies the difference between the Jacobian matrices of the current mesh configuration and the RJMs as shown below:

$$\psi^{\mathrm{R}} = \sum_i \sum_j \frac{\left|\mathbf{J}_{ji} - \mathbf{J}^{\mathrm{R}}_{ji}\right|_F^2}{\left|\mathbf{J}_{ji}\right|^2 A_j / A^{\mathrm{R}}_{ji}}, \; i \in \{i|V_i \in \mathscr{V}\}, \; j \in \{j|F_j \in \mathscr{F}(V_i)\} \qquad (6)$$

where $\mathscr{V}$ is the set of all mesh vertices and $A_{ji}{}^{\mathrm{R}}$ is the area of the triangle formed by edge vectors, $(\mathbf{e}_p{}^{\mathrm{R}})_i$ and $(\mathbf{e}_q{}^{\mathrm{R}})_i$. Note that, similar to the objective function for local optimization, the objective function includes a barrier term, A_j, in the denominator in the form of the triangle area to prevent mesh invalidity. Since the Jacobian matrix and the RJM are formed from the mesh edges and the reference edges, respectively, optimization of Ψ^{R} makes the edges of the final mesh as close as possible to their respective reference edge vectors.

As with the condition number based optimization, the global objective function, Ψ^{R} is minimized by iteratively minimizing a local component of the global function at each mesh vertex. The local component of the global objective function that involves the real and reference positions of V_i is given as:

$$\tilde{\psi}_i^{\mathrm{R}} = \sum_j \sum_k \frac{\left\| \mathbf{J}_{jk} - \mathbf{J}_{jk}^{\mathrm{R}} \right\|^2}{\left|\mathbf{J}_{jk}\right|^2 A_j / A_{jk}^{\mathrm{R}}}, \ j \in \left\{ j | F_j \in \mathscr{F}(V_i) \right\},$$
$$k \in \left\{ k | V_k \in \mathscr{V}\left(F_j\right) \cap \mathscr{V}(\mathscr{E}(V_i)) \right\}$$

In the above expression, the outer sum is over all faces connected to the vertex and the inner sum is over all vertices of a face that include V_i itself, or are edge-connected to V_i.

4 Results

Figure 5 shows a simple example to illustrate the effects of a condition number optimization (CNO) and reference Jacobian based optimization (RJO) on a nonplanar surface mesh. Figure 5a shows the original pyramid-shaped mesh on which the two optimization techniques are applied. Figure 5b shows the effect of optimizing the CN objective function and Figure 5c shows the effect of optimizing the RJ objective function. In both cases, the apex vertex lies on the left lateral surface of the original pyramid. It can be seen that CNO improves the shapes of the triangles more than RJO. On the other hand, RJO results in lesser movement of the apex vertex from its original position.

Figure 6a shows the polygonal mesh of a pig, and Fig. 6b, c show the results of CNO and RJO on the mesh, respectively. It is again clear from the example that CNO improves the shape of mesh elements more than RJO, but it also causes much more movement of the vertices. In particular, note that CNO destroys much of the anisotropy in the midsection of the pig and smooths away the local refinement around the pig's mouth while RJO preserves these characteristics of the mesh. Table 1 shows the histograms of the *normalized average condition number* of elements before and after

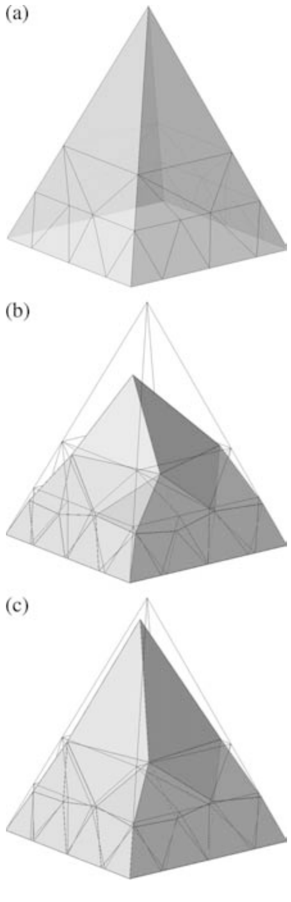

Fig. 5 **a** Original mesh. **b** Mesh optimized with condition number objective function, **c** Optimized with reference Jacobian objective function. Note that, in both cases, the apex vertex is on the lateral surface of the original pyramid

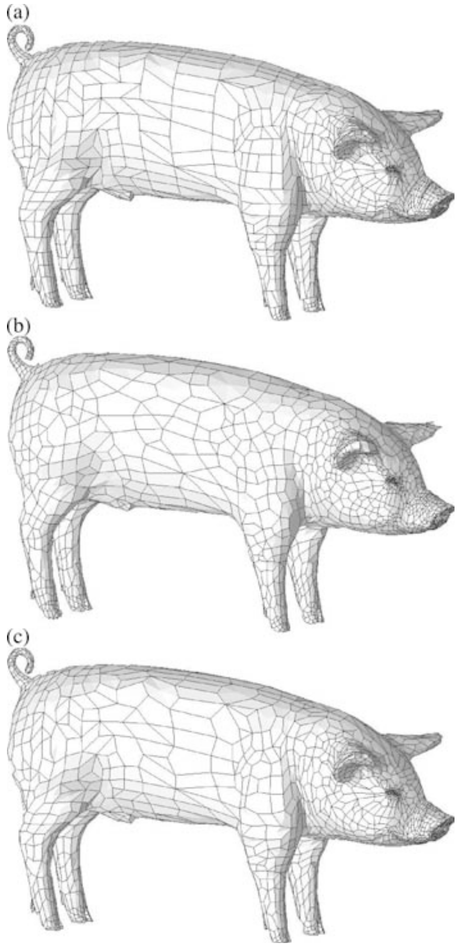

Fig. 6 **a** Mesh of pig with anisotropy and local refinement. **b** Mesh optimized with global condition number objective function. **c** Mesh optimized with reference Jacobian objective function

Table 1 Histograms of normalized average condition number of elements in original and optimized polygonal meshes of a pig (Fig. 6)

$\bar{\mathscr{K}}$	Original	CNO	RJO
1.0–1.5	1,100	2,668	1,768
1.5–2.0	1,017	304	855
2.0–3.0	736	49	364
3.0–4.0	113	5	31
4.0–5.0	25	1	7
5.0–7.5	21	0	3
7.5–10.0	11	1	0
10.0–15.0	3	1	1
15.0	3	0	0

Table 2 Quantitative measures of the change in the mesh and discrete surface characteristics for CNO and RJO for polygonal mesh of a pig (Fig. 6); all values, except the change in normals, are presented as a percentage of the problem size

Measure	CNO	RJO
Average change in normals	10.7°	4.1°
Hausdorff distance	0.3%	0.1%
Maximum vertex movement	7.8%	2.6%
Average vertex movement	1.0%	0.2%

the two types of optimization. The normalized average condition number for an element is defined as the mean of the condition numbers at the vertices of an element, normalized so that a regular polygon will produce a value of 1.

Table 2 shows various quantities computed to measure the change in the meshes and the discrete surfaces using the two methods of optimization. In the table, the normalized Hausdorff distance is computed by finding the minimum distance from each vertex of the original mesh to the new mesh, taking the maximum of these distances [23, 24], and then normalizing it by the problem size. The problem size is defined as the maximum length of the domain along the three coordinate directions. The difference between discrete normals is the angle between the normal vector of quadrics fitted to the neighborhood of a vertex at its old and new locations [18, 19]. The maximum vertex movement is the maximum distance traveled by any vertex from its original position and the average vertex movement is the mean of the distance traveled by all vertices from their original positions; these are also normalized by the problem size.

Finally, a complex mesh of an "Moai" statue is presented in Fig. 7 to illustrate the effectiveness of this procedure on large surface meshes. The original mesh for this model was obtained from the Web sites of Belyaev and Ohtake[2] and then converted into a polygonal mesh along with the application of some compression below the neck and stretching at the chest. The modified mesh (Fig. 7a) was used to obtain the optimized meshes shown in the example. CNO resulted in the mesh shown in Fig. 7b and RJO yielded the mesh shown in Fig. 7c.

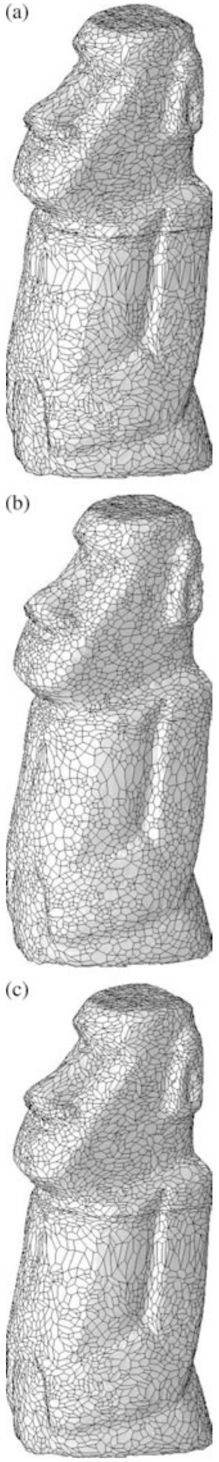

Fig. 7 a Polygonal mesh of the Moai model (courtesy of Belyaev and Ohtake). **b** Mesh optimized with CN objective function. **c** Mesh optimized with RJ objective function

As with the pig, it can be seen in the Moai mesh that the CNO improves the mesh considerably but eliminates some significant features in the mesh (particularly the refinement) while the mesh obtained by RJO preserves

[2] http://www.mpi-sb.mpg.de/~belyaev/soft/ply/gallery.html

Table 3 Histograms of normalized average condition number in original and optimized meshes for Moai mesh (Fig. 7)

$\bar{\mathcal{K}}$	Original	CNO	RJO
1.0–1.5	3,250	7,029	5,315
1.5–2.0	2,155	464	1,722
2.0–3.0	1,471	67	456
3.0–4.0	383	8	58
4.0–5.0	159	3	15
5.0–7.5	111	2	5
7.5–10.0	32	0	1
10.0–15.0	7	1	1
15.0	6	0	1

Table 4 Quantitative measures of the change in the mesh and discrete surface characteristics for CNO and RJO for Moai mesh (Fig. 7); all values except the change in normals are presented as a percentage of the problem size

Measure	CNO	RJO
Average change in normals	5.1°	2.2°
Hausdorff distance	0.20%	0.09%
Maximum vertex movement	3.7%	2.1%
Average vertex movement	0.66%	0.19%

these features. The condition number histograms for the three meshes are presented in Table 3 and the measures for change in surface characteristics are presented in Table 4.

With respect to execution time, RJO is normally faster than CNO since the former causes lesser node movement. Timing statistics collected for the above test cases and other examples indicate that CNO takes 10%–50% longer than RJO.

5 Conclusions

A procedure was presented to improve the quality of complex polygonal surface meshes without an underlying smooth surface using numerical optimization. The optimization is designed to improve the quality of the mesh faces without distorting the discrete surface too much. The vertices are kept on the original surface mesh using movement in local parametric spaces of mesh faces. Two methods were proposed for improving the quality of the surface mesh. The first method improved the quality of mesh elements as much as possible by minimizing a global condition number objective function by local iteration. The second method was the two-stage reference Jacobian matrix or RJM-based method, which improved the mesh quality as well as minimized the movement of vertices from their original locations.

The procedure has been successfully tested on a number of complex polygonal surface meshes. Several quantitative measures were presented to show that both types of optimizations do not distort the surface much.

Acknowledgements The work of the authors was performed at Los Alamos National Laboratory operated by the University of California for the US Department of Energy under contract W-7405-ENG-36. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

References

1. Field DA (1988) Laplacian smoothing and Delaunay triangulations. Commun Appl Numerical Methods 4:709–712
2. Freitag L, Jones M, Plassmann P (1995) An efficient parallel algorithm for mesh smoothing. In: Proceedings of the 4th international meshing roundtable (4IMR), Albuquerque, New Mexico, October 1995. Sandia National Laboratories, Sandia report SAND 95-2130, pp 47–58
3. Zavattieri PD, Dari EA, Buscaglia GC (1996) Optimization strategies in unstructured mesh generation. Int J Numerical Methods Eng 39:2055–2071
4. Knupp P (1998) Winslow smoothing on two-dimensional unstructured meshes. In: Proceedings of the 7th international meshing roundtable (7IMR), Dearborn, Michigan, October 1998. Sandia National Laboratories, Sandia report SAND 98-2250, pp 449–457
5. Cannan SA, Tristano JR, Staten ML (1998) An approach to combined laplacian and optimization-based smoothing for triangular, quadrilateral and quad-dominant meshes. In: Proceedings of the 7th international meshing roundtable (7IMR), Dearborn, Michigan, October 1998. Sandia National Laboratories, Sandia report SAND 98-2250, pp 479–494
6. Zhou T, Shimada K (2000) An angle-based approach to two-dimensional mesh smoothing. In: Proceedings of 9th international meshing roundtable (9IMR), New Orleans, Louisiana, October 2000. Sandia National Laboratories, Sandia report SAND 2000-2207, pp 373–384
7. Garimella RV, Shashkov MJ, Knupp PM (2004) Triangular and quadrilateral surface mesh quality optimization using local parameterization. Comput Methods Appl Mech Eng 193(9–11):913–928
8. Kuriyama S (1999) Diffusive smoothing of polygonal meshes with bias and tension control. Vis Comput 15(10):509–518
9. Jose RS, Alberola C, Ruiz J (2001) Reshaping polygonal meshes with smooth normals extracted from ultrasound volume data: an optimization approach. Proc SPIE 4325:462–472
10. Garimella RV, Shashkov MJ, Knupp PM (2002) Optimization of surface mesh quality using local parameterization. In: Proceedings of the 11th international meshing roundtable (11IMR), Ithaca, New York, September 2002. Sandia National Laboratories, pp 41–52, http://cnls.lanl.gov/~shashkov
11. Giannakoglou KC, Chaviaropoulos P, Papailiou KD (1996) Boundary-fitted parametrization of unstructured grids on arbitrary surfaces. Adv Eng Soft 27:41–49
12. Floater MS (1997) Parametrization and smooth approximation of surface triangulations. Comput Aid Geometric Des 14:231–250
13. Sheffer A, de Sturler E (2001) Parameterization of faceted surfaces for meshing using angle-based flattening. Eng Comput 17(3):326–337
14. Desbrun M, Meyer M, Alliez P (2002) Intrinsic parameterizations of surface meshes. In: Proceedings of the Eurographics 2002 conference, Saarbrücken, Germany, September 2002
15. Guskov I (2002) An anisotropic mesh parameterization scheme. In: Proceedings of the 11th international meshing roundtable (11IMR), Ithaca, New York, September 2002. Sandia National Laboratories, pp 325–332
16. Zienkiewicz OC, Taylor RL (1987) The finite element method, vol 1, 4th edn. McGraw-Hill, New York

17. Hughes TJR (1987) The finite element method: linear static and dynamic finite element analysis. Prentice Hall, Englewood Cliffs, New Jersey
18. McIvor AM, Valkenburg RJ (1997) A comparison of local surface geometry estimation methods. Machine Vis Appl 10:17–26
19. Petitjean S (2002) A survey of methods for recovering quadrics in triangle meshes. ACM Comput Surveys 34(2):211–262
20. Knupp PM (2000) Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I—a framework for surface mesh optimization. Int J Numerical Methods Eng 48(3):401–420
21. Shashkov MJ, Knupp PM (2001) Optimization-based reference-matrix rezone strategies for arbitrary Lagrangian-Eulerian methods on unstructured grids. In: Proceedings of the 10th anniversary international meshing roundtable (10IMR), Newport Beach, California, October 2001. Sandia National Laboratories, Sandia report SAND 2001-2976C, pp 167–176
22. Knupp PM, Margolin LG, Shashkov MJ (2002) Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. J Comput Phys 176:93–128
23. Cignoni P, Rocchini C, Scopigno R (1998) Metro: measuring error on simplified surfaces. Comput Graph Forum 17(2): 167–174
24. Huttenlocher DP, Klanderman GA, Rucklidge WJ (1993) Comparing images using the Hausdorff distance. IEEE Trans Pattern Anal Mach Intell 15(3):850–863