A back-tracking procedure for optimization of simplex meshes

Joaquim B. Cavalcante-Neto^{1,*,†}, Luiz Fernando Martha^{2,‡}, Paul A. Wawrzynek^{3,§} and Anthony R. Ingraffea^{3,¶}

¹Computer Graphics Group (CRAb), Federal University of Ceara (UFC), Brazil ²Computer Graphics Technology Group (Tecgraf), Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil ³Cornell Fracture Group (CFG), Cornell University, U.S.A.

SUMMARY

This work presents a back-tracking procedure for the optimization of meshes. It can be applied to all ill-shaped elements in a mesh in order to improve quality in mesh optimization. Basically, the original mesh is reconstructed in regions around the ill-shaped elements by means of a set of deletion and reconstruction operations based on visibility tests. The back-tracking procedure fixes problems related to the quality of generated meshes in general and can be applied in two or three dimensions. The procedure works by deleting elements ranked below a predefined shape quality measure. It is important to mention that, although the presented procedure was devised for advancing-front algorithms, it can also be used in meshes generated by different approaches, such as Delaunay and others, requiring no modification. The proposed procedure is applicable to simplex elements (triangles and tetrahedra), although ideally it could be extended to other types of elements. Examples of improved meshes using the back-tracking procedure are presented, in which the quality of these meshes is assessed in order to validate the procedure proposed in this work. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: mesh generation; mesh optimization; adaptivity

1. INTRODUCTION

This work presents a back-tracking procedure for the optimization of meshes in 2D and 3D. The back-tracking procedure can be applied to all ill-shaped elements in a mesh, in order to improve quality in mesh optimization.

[†]E-mail: joaquimb@lia.ufc.br

Contract/grant sponsor: CNPq; contract/grant number: 307148/2004-1

Received 21 December 2004 Revised 17 March 2005 Accepted 22 March 2005

Copyright © 2005 John Wiley & Sons, Ltd.

^{*}Correspondence to: Joaquim Bento Cavalcante-Neto, Department of Computing (DC), Federal University of Ceara (UFC), Campus do Pici, Bloco 910, 60455-760, Fortaleza-CE, Brazil.

[‡]E-mail: Ifm@tecgraf.puc-rio.br

[§]E-mail: wash@fac.cfg.cornell.edu

[¶]E-mail: ari1@cornell.edu

There are several techniques described in the literature that address, in one way or another, the problem of mesh optimization. These techniques represent a wide variety of solutions, from simple to very complicated ones, but the general idea is to somehow improve the quality of the final mesh generated.

One example of a technique for mesh optimization, is the so-called *mesh smoothing*, which is probably the most used and one of the most effective techniques found in the literature. It consists of relocating node co-ordinates to improve mesh quality. Mesh smoothing usually increases the quality of the mesh as well as smoothing the transition among elements of various sizes. In general, there are two common methods used in mesh smoothing: Laplacian smoothing [1-3] and optimal smoothing [4]. There are also variations of these methods, such as the weighted Laplacian smoothing approach, in which the smoothing is performed by relocating interior nodes within a patch, using a weight called the *relaxation coefficient* [5]. This method can be very effective and bring significant improvement, especially in three dimensions (3D) [6]. The weight is necessary because the smoothing truly improves the quality in two dimensions (2D), but in 3D this is more difficult to ensure. An additional problem is that the relocation of interior nodes can occasionally result in invalid elements with negative volumes. For this reason, additional procedures to check if the relocation has maintained the integrity of the mesh are necessary.

Another example of a commonly used strategy for mesh optimization is a technique that consists in swapping entities in a mesh [3]. The entities are usually edges or faces. This technique is also very effective, but since only local changes are performed, affecting only local connectivity, elements that are very badly shaped, especially in 3D, may not be sufficiently improved.

The idea of deleting poorly shaped elements or regions to improve mesh quality has also been recently addressed in the literature. For instance, the work by [7] uses local mesh regeneration based on the deletion of sliver tetrahedra in a post-processing step. The procedure proposed in the present work has the same purpose but uses a different strategy. Rassineux [8] also optimizes the mesh by reconstructing sub-volumes that are obtained with the deletion of a group of tetrahedra.

The technique proposed here, called *back-tracking*, is a new procedure devised to improve mesh quality in 2D or 3D. It is applied to all ill-shaped elements in a mesh, reconstructing the original mesh in regions around these elements. In short, the back-tracking procedure is based on the following steps. The initial local region for remeshing is identified by grouping a given ill-shaped element with adjacent elements that share edges (in 2D) or faces and edges (in 3D) with it. Then, visibility tests related to the centroid of this region are performed, in order to verify whether this centroid is visible from all nodes on the boundary of the region. If any boundary node is not visible from the centroid, the region is expanded with elements that share edges (in 2D) or faces and edges (in 3D) with the elements that contain the non-visible nodes. This procedure is repeated for this expanded region and continues until all nodes on the boundary of the selected region are visible from the centroid. After a visible region is found, new elements are generated, connecting the elements on the boundary of the region to the centroid. Since the region is star-shaped, usually close to a convex form, the remeshing results in well-shaped elements.

The back-tracking procedure fixes problems related to mesh quality in general. It can be applied in 2D or 3D, although it can be shown that in 2D such problems are more less likely to occur. The procedure works by deleting elements ranked below a predefined shape quality

measure. It is important to mention that, although the presented procedure was devised for advancing-front algorithms, it can also be used for meshes generated by different approaches, such as Delaunay and others, requiring no modification. The proposed procedure is applicable to simplex elements (triangles and tetrahedra), although ideally it could be extended to other types of elements.

The remainder of this work is organized as follows. Section 2 describes the overall backtracking strategy and its application in 2D and 3D. Section 3 shows validation studies that attest to the robustness of the proposed technique. Finally, in Section 4 some conclusions are presented.

2. BACK-TRACKING STRATEGY

This section describes the general strategy of the back-tracking procedure and its application in 2D and 3D. In each case, specific details are discussed.

2.1. General description

The idea of the strategy is to delete an ill-shaped element, generate a region around it and try to insert a new node in the centroid of this region, in order to generate better-shaped new elements. If this region does not have a kernel, which is a core where any point is visible through a straight line from all of its vertices to the centroid, then this region is expanded and the process is repeated, until a kernel is found. This process tries to find a non-meshed region, represented by a star-shaped polygon in 2D (or polyhedron in 3D), relative to its centroid, where new elements with better shape can be formed by the centroid and its vertices.

Let \mathbf{E}_k be an ill-shaped element of the generated mesh and $\mathbf{R}(\mathbf{E}_k)$ the local region for remeshing in the neighbourhood of \mathbf{E}_k , which is identified by grouping the ill-shaped element with adjacent elements that share edges (in 2D) or faces and edges (in 3D) with it. Let $\mathbf{M}(\mathbf{E})$ be the set of all elements in the current mesh and $\mathbf{D}(\mathbf{E}_k)$ the set of elements deleted during the back-tracking process, if necessary. Finally, let $\mathbf{F}(\mathbf{E}_k)$ be the set of new elements generated for the local region $\mathbf{R}(\mathbf{E}_k)$, when a kernel for this region is found. The procedure used to transform an ill-shaped region into one with a visible kernel is as follows:

- The boundary of the local region $\mathbf{R}(\mathbf{E}_k)$ related to the current ill-shaped element \mathbf{E}_k is identified.
- A visibility test is performed. It consists of computing the co-ordinates of the centre of this region $\mathbf{R}(\mathbf{E}_k)$ and counting the number of intersections that would occur, for each of the elements in this region, if lines were drawn from the centre to all of the region's vertices.
- If there is at least one intersection for any of the region's elements, then this region must be modified. This is done by removing the element that has the highest number of intersections and its vicinity, if necessary. These elements form the set $D(E_k)$ for the current region. This generates a new region $R(E_k)$. Also, the set of all elements in the current mesh M(E) must be updated to account for the elements deleted.
- This process is repeated until the centre of the region $\mathbf{R}(\mathbf{E}_k)$ is visible from all vertices in this region. When this happens, the set of new elements $\mathbf{F}(\mathbf{E}_k)$ is generated for the

region $\mathbf{R}(\mathbf{E}_k)$ by connecting the centre to all of the region's vertices. The set of all elements in the current mesh $\mathbf{M}(\mathbf{E})$ must be updated to account for these new elements $\mathbf{F}(\mathbf{E}_k)$ generated.

It is possible that the process of finding a star-shaped region may fail if elements to be removed are attached to the boundary and therefore cannot be deleted. If this occurs, internal alternative elements are deleted instead and the process is restarted. If the region is still not meshable, the procedure fails and terminates, and the original mesh for this particular region, which was 'saved' in the beginning of the process, is restored. In principle, it is possible to create a boundary input mesh that forces this failure, but this has not yet been observed.

Another important observation is that the back-tracking procedure is not guaranteed to always improve mesh quality, especially in 3D, since the insertion of points inside starshaped polyhedra can, in some cases, worsen the mesh. If this happens, the original mesh for the current region is restored. The approach used in this work to compare the relative quality of two meshes is to check if all new elements are better shaped than the worst element in the original mesh, considering the elements deleted during the back-tracking for this region. Any quality metric can be used for the comparison and altogether different techniques can be employed to compare new and old meshes. In the vast majority of the tests, a significant improvement in mesh quality was observed. It is possible that, in some cases, the back-tracking procedure may create situations where many elements share the new centroid node, thus the angles at this node would be relatively small and possibly unacceptable. If an element with small angles is a concern, then a quality metric that is sensitive to angles should be chosen. In practice, however, such situations have rarely been observed.

A general algorithm for the back-tracking procedure, which consists of all steps previously described, is as follows:

- Determine region $\mathbf{R}(\mathbf{E}_k)$ for the ill-shaped element \mathbf{E}_k .
- Perform visibility tests for region $\mathbf{R}(\mathbf{E}_k)$.
- If there are intersections, update region $\mathbf{R}(\mathbf{E}_k)$ by $\mathbf{R}(\mathbf{E}_k)_{\text{new}} = \mathbf{R}(\mathbf{E}_k)_{\text{old}} \mathbf{D}(\mathbf{E}_k)$. Also, update the current mesh $\mathbf{M}(\mathbf{E})$ by $\mathbf{M}(\mathbf{E})_{\text{new}} = \mathbf{M}(\mathbf{E})_{\text{old}} \mathbf{D}(\mathbf{E}_k)$.
- Repeat the process until a kernel is found. When this happens, find the set of new elements $F(E_k)$ and update M(E) by $M(E)_{new} = M(E)_{old} + F(E_k)$.

The overall mesh optimization, for a given mesh, terminates when all ill-shaped elements of the original mesh are examined. Some ill-shaped elements can even disappear before they are examined, because they are deleted by the back-tracking procedure applied to neighbouring elements. Every ill-shaped element is evaluated only once, since, after the back-tracking procedure, its quality is improved (or maintained, in case the original mesh is restored for a particular region).

2.2. Back-tracking in two and three dimensions

The application of the back-tracking strategy for problems in 2D and 3D follows the general description, with some specific details for each case. For instance, in 2D, the entities involved in the strategy are faces (triangles), while in 3D they are tetrahedra, with their respective roles.

In 2D, the back-tracking procedure consists of deleting an element classified as poorly shaped and a group of elements in its vicinity. These elements are faces (triangles). The classification of a 'bad' face is based on a metric for triangular shapes. Elements with a value below a threshold are classified as poorly shaped.

The purpose of the procedure is to delete a 'bad' triangle in order to create a local region, in this case a polygon, that can be remeshed with better-shaped triangles. This is illustrated in Figure 1, which shows a 'bad' triangle and the polygon associated with it.

After the creation of the local polygon, an attempt is made to generate triangles by inserting a new internal node in the polygon's centre, as is also shown in Figure 1. If this does not work, the back-tracking procedure is employed.

In 3D, the back-tracking strategy is similar to the two-dimensional case, the only differences being that the elements are now tetrahedra and the local region to be remeshed is now represented by a polyhedron. It is important to mention, however, that in 3D it is usually much more difficult to find the kernel of the polyhedron, and sometimes it may not even exist. For this reason, it is important to 'save' the original configuration of the mesh for a particular ill-shaped element, in case the kernel cannot be found after a number of steps and thus nothing can be done to improve the quality of this element. The number of steps is defined empirically and it is important to avoid an undesirable number of elements adjacent to the centre of the region that might occur. In this work, a maximum of 10 steps was used. The same caution can be taken in 2D, but usually the process converges well below the maximum number of steps defined.

Figure 2 illustrates the transformation of a polyhedron into a star-shaped one, relative to its centroid, after an 'intersected' element is deleted. In this example, a line drawn from node b to the centroid intersects face *acd*. This results in the removal of the tetrahedron formed by nodes a, b, c and d. The resulting polyhedron is shown at the centre of Figure 2. This figure also shows the tetrahedral elements generated after the insertion of a node at the centroid of the new polyhedron.

The algorithm for the two (three)-dimensional case is similar to the general one, where $\mathbf{R}(\mathbf{E}_k)$ is represented by the polygon (polyhedron) $\mathbf{P}(\mathbf{T}_k)$, \mathbf{E}_k is represented by the ill-shaped triangle (tetrahedron) \mathbf{T}_k , $\mathbf{M}(\mathbf{E})$ is represented by the set of triangles (tetrahedra) of the mesh $\mathbf{M}(\mathbf{T})$, $\mathbf{D}(\mathbf{E}_k)$ is represented by the set of deleted triangles (tetrahedra) $\mathbf{D}(\mathbf{T}_k)$, and $\mathbf{F}(\mathbf{E}_k)$ is represented by the set of new triangles (tetrahedra) $\mathbf{F}(\mathbf{T}_k)$.



Figure 1. Two-dimensional back-tracking procedure around a 'bad' face.

Commun. Numer. Meth. Engng 2005; 21:711-722

Copyright © 2005 John Wiley & Sons, Ltd.



Figure 2. Transformation of a polyhedron into a star-shaped one.

3. VALIDATION STUDIES

In this section, the quality measures used in this work are discussed. Also, the quality of the meshes improved by the back-tracking technique is evaluated.

3.1. Quality measures

In the proposed back-tracking technique, mesh improvement implies that element-shape quality measures are required. These measures will define a threshold used to determine if an element is considered ill-shaped. This issue is well represented in the literature [5, 9-14], with measures of all kinds.

Considering that, in 3D, it is more difficult to ensure that high-quality meshes are generated, the examples shown are three-dimensional ones, in order to attest the efficiency of the proposed technique. For this reason, the metric adopted in this work is a normalized ratio between the root mean square of the lengths of the edges of a tetrahedron, represented by

$$S_{\rm rms} = \sqrt{\frac{1}{6} \sum_{i=0}^{5} S_i^2}$$
(1)

where S_i is the length of and edge, and the volume V of the tetrahedron [12]:

$$\gamma = \frac{S_{\rm rms}^3}{V} \tag{2}$$

This metric generates a good quality measure and is computationally efficient. The range of valid values varies from $[1, \infty)$ [5, 11], and the optimal value for the regular tetrahedron is approximately 8.5. For each element, this metric is evaluated and, if it is outside a predefined range, the element is classified as ill-shaped. Lower and upper bounds of this range are defined empirically, based on experiments and observations. In this work, the lower bound value is 5.0 and the upper bound is the 'optimal' metric value of 8.5 multiplied by a factor of 30.

It is important to mention, however, that any quality measure can be used, since all that the strategy requires is some metric to indicate if a particular element is ill-shaped or not. Moreover, the same concept applies to two-dimensional problems, where a metric should be used to define the ill-shaped elements of the mesh.

Copyright © 2005 John Wiley & Sons, Ltd.

For the evaluation of the final quality of the mesh, after the application of the back-tracking technique, a different metric was adopted. The metric α is used, which is defined by

$$\alpha = \frac{3R_{\rm i}}{R_{\rm c}} \tag{3}$$

where R_i and R_c are the radii of the inscribed and circumscribed sphere, respectively. This metric is equal to 1.0 for the regular tetrahedron and sliver tetrahedra generally have values lower than 0.1. The α metric, instead of the γ metric used to identify the ill-shaped elements, is adopted in the evaluation of the quality of the final meshes because its interpretation is more intuitive and it is more widely referenced in the literature. In addition, from the work by Reference [13] it is known that both metrics (γ and α) are equivalent.

3.2. Three-dimensional examples

Two illustrative examples are presented here: a portion of a housing and a portion of a spiralbevel gear. These are shown in Figures 3–6. Both examples contain small surface cracks. The cracks are idealized as having no volume, that is, the surfaces representing the two sides of a crack are topologically distinct but geometrically coincident. The existence of cracks demonstrates that the proposed technique can handle complex constraints in the generated mesh.

Figures 7 and 8 show histograms of the number of generated elements in various ranges of α for all models, comparing the back-tracking procedure with other mesh optimization schemes, such as Laplacian smoothing [5] and face swapping [3]. Table I shows some statistics related to α for the examples, comparing the optimized mesh generated by the back-tracking procedure with the original one.

The histograms show that the majority of the elements are located in the range [0.7, 0.8], which represents well-shaped elements, even before the application of the back-tracking



Figure 3. Mesh of the housing example.

Commun. Numer. Meth. Engng 2005; 21:711-722

Copyright © 2005 John Wiley & Sons, Ltd.



Figure 4. Mesh detail showing an embedded crack in the housing example: crack region.



Figure 5. Mesh of the gear example.

Copyright © 2005 John Wiley & Sons, Ltd.

Commun. Numer. Meth. Engng 2005; 21:711-722



Figure 6. Mesh detail showing an embedded crack in the gear example: crack region.



Figure 7. Element quality histogram of the housing example.

procedure. These meshes were generated by an advancing-front-based algorithm [6] which already takes care of quality during mesh generation. However, the number of elements in this range increases after the application of the back-tracking procedure in an order of 2-4%. Although this percentage might seem small, it can represent a large number of elements in big



Figure 8. Element quality histogram of the gear example.

Table I. Statistical values related to mesh quality in all examples.

| Example | Histogram | # Elements | $\alpha_{\rm avg}$ | α_{min} | α_{max} |
|---------|---------------|------------|--------------------|----------------|----------------|
| Housing | Original | 16463 | 0.675 | 0.025 | 0.729 |
| Housing | Back-tracking | 17043 | 0.696 | 0.023 | 0.740 |
| Gear | Original | 17386 | 0.684 | 0.025 | 0.738 |
| Gear | Back-tracking | 16990 | 0.699 | 0.033 | 0.742 |

meshes. This result is an indication that this percentage could increase in cases of meshes containing a larger number of ill-shaped elements before the application of the back-tracking. Regarding the poorly shaped elements initially there was a significant number of elements (1.76– 3.77%) with α values lower than 0.1, which represent undesirable elements. After the application of the back-tracking procedure, the number of poorly shaped elements drops significantly (0.49–0.90%).

Compared with the other mesh optimization schemes, the back-tracking procedure generates more elements in the range [0.7, 0.8] than any other scheme tested. Related to the smoothing technique, this better result is probably due to the fact that smoothing relocates nodes within a patch, resulting in a procedure based on local modifications that, although it can be generally very effective, does not have the possibility of expansion performed by the back-tracking. In relation to the face swapping, the local modifications also performed by this technique probably do not allow the same expansion necessary in some cases. It is important to observe, however, that face swapping can get better results in the overall distribution, if the other ranges are considered, since it can make elements 'jump' more easily to a neighbour range, whereas the back-tracking can move more elements from distant ranges (for instance, from the range [0, 0.1] to the range [0.7, 0.8]), probably again because of the possibility of expansion. These

OPTIMIZATION OF SIMPLEX MESHES

| Example | Procedure | # Entities | Time (s) |
|---------|---------------|------------|----------|
| Housing | Face swapping | 404 | 5 |
| Housing | Back-tracking | 140 | 13 |
| Gear | Face swapping | 487 | 7 |
| Gear | Back-tracking | 189 | 16 |

Table II. Speed measurements in all examples.

results attest that the back-tracking can really improve mesh quality in a very effective way and in some situations can be better than other schemes.

Table II shows some speed measurements, in seconds, for the examples, comparing the back-tracking and the face swapping procedures. In both procedures the number of entities affected is shown, which are tetrahedra to be deleted (in back-tracking) and faces to be swapped (in face swapping). Although the important information in this table is the time spent to improve the mesh and considering that these entities are different (tetrahedra and faces), this information is relevant to give an idea of how many entities had to be modified in order to improve the mesh. The times were measured using a 1.333 GHz AMD Athlon(tm) processor. It can be seen from the table that the time spent in face swapping is practically non-existent, as expected, even considering that around 400 faces had to be swapped in each example. The time spent in the back-tracking is higher, as expected, but it is still very low, even considering that around 200 tetrahedra had to be deleted tetrahedron is usually very local, small and then the operations to be performed in this region are not expensive. This observation attests that, regarding time consumption, the back-tracking can be used in practical applications.

4. CONCLUSIONS

A back-tracking procedure for the optimization of meshes in 2D and 3D was presented. This procedure can be applied to all ill-shaped elements in a mesh in order to improve quality in mesh optimization. The proposed procedure is applicable to simplex elements (triangles and tetrahedra), although ideally it could be extended to other types of elements.

The strategy of the procedure consists in deleting an identified ill-shaped element, generating a new region around this element and trying to insert a new node in the centroid of this region, in order to generate better-shaped new elements. If this region does not have a kernel, which is a core where any point is visible through a straight line from all of its vertices to the centroid, then this region must be expanded and the process repeated until a kernel is found. This process tries to find a non-meshed region, represented by a star-shaped polygon in 2D (or polyhedron in 3D), relative to its centroid, where new elements with better shape can be formed by the centroid and its vertices.

A number of realistic examples was shown to demonstrate the quality of the optimized meshes. It was shown that only a small percentage (0.49-0.90%) of the total number of elements are still poorly shaped, after the application of the back-tracking procedure. It was also shown that the presented procedure improves the percentage of elements considered of very good shape.

ACKNOWLEDGEMENTS

The first author acknowledges the support from Brazilian agency CNPq through the grant 307148/2004-1.

REFERENCES

- 1. Field D. Laplacian smoothing and Delaunay triangulation. Communications in Applied Numerical Methods 1988; 4(1):709-712.
- 2. Hansbo P. Generalized laplacian smoothing of unstructured grids. *Communications in Numerical Methods in Engineering* 1995; **11**(1):455-464.
- 3. Freitag L, Ollivier-Gooch C. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering* 1997; **40**(1):3979–4002.
- Borouchaki H, George P. Aspects of 2-d Delaunay mesh generation. International Journal for Numerical Methods in Engineering 1997; 40(1):1957–1975.
- 5. Lewis R, Zheng Y, Gethin D. Three-dimensional unstructured mesh generation. Part 3: volume meshes. Computer Methods in Applied Mechanics 1996; **134**(1):285-310.
- 6. Cavalcante-Neto J, Wawrzynek P, Carvalho M, Martha L, Ingraffea A. An algorithm for three-dimensional mesh generation for arbitrary regions with cracks. *Engineering with Computers* 2001; **17**(1):75–91.
- 7. Chan C, Anastasiou K. An automatic tetrahedral mesh generation scheme by the advancing front method. *Communications in Numerical Methods in Engineering* 1997; **13**(1):33–46.
- Rassineux A. Generation and optimization of tetrahedral meshes by advancing front technique. *International Journal for Numerical Methods in Engineering* 1998; 41(1):651–674.
- 9. Cavendish J, Field D, Frey W. An approach to automatic three-dimensional finite element mesh generation. International Journal for Numerical Methods in Engineering 1995; 21(1):329-347.
- Joe B. Delaunay versus max-min solid angle triangulation for three-dimensional mesh generation. International Journal for Numerical Methods in Engineering 1991; 31(1):987–997.
- 11. Parthasarathy V, Graichen C, Hathaway A. A comparison of tetrahedron quality measures. *Finite Element Analysis and Design* 1993; **15**(1):255–261.
- 12. Weatherill N, Hassan O. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering* 1994; **37**(1): 2005–2039.
- 13. Liu A, Joe B. Relationship between tetrahedron shape measures. BIT 1994; 34(1):268-287.
- 14. Freitag L, Knupp P. Tetrahedral element shape optimization via the Jacobian determinate and condition number. 8th International Mesh Roundtable 1999; 247–258.