ELSEVIER

Journal of Biomechanics 34 (2001) 815–819

JOURNAL OF BIOMECHANICS

www.elsevier.com/locate/jbiomech
www.JBiomech.com

Technical note

Merging of intersecting triangulations for finite element modeling

Juan R. Cebral[a,*], Rainald Löhner[a], Peter L. Choyke[b], Peter J. Yim[b]

[a] *School of Computational Sciences, George Mason University, 4400 University Drive, MS 4C7, Fairfax, VA 22030, USA*
[b] *Imaging Sciences Program, National Institutes of Health, Building 10, Room 1C660, Bethesda, MD 20892, USA*

Accepted 23 January 2001

Abstract

Surface mesh generation over intersecting triangulations is a problem common to many branches of biomechanics. A new strategy for merging intersecting triangulations is described. The basis of the method is that object surfaces are represented as the zero-level iso-surface of the distance-to-surface function defined on a background grid. Thus, the triangulation of intersecting objects reduces to the extraction of an iso-surface from an unstructured grid. In a first step, a regular background mesh is constructed. For each point of the background grid, the closest distance to the surface of each object is computed. Background points are then classified as external or internal by checking the direction of the surface normal at the closest location and assigned a positive or negative distance, respectively. Finally, the zero-level iso-surface is constructed. This is the final triangulation of the intersecting objects. The overall accuracy is enhanced by adaptive refinement of the background grid elements. The resulting surface models are used as support surfaces to generate three-dimensional grids for finite element analysis. The algorithms are demonstrated by merging arterial branches independently reconstructed from contrast-enhanced magnetic resonance images and by adding extra features such as vascular stents. Although the methodology is presented in the context of finite element analysis of blood flow, the algorithms are general and can be applied in other areas as well. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Surface intersection; Grid generation; Finite element analysis; Hemodynamics; Magnetic resonance images; Carotid artery

1. Introduction

Accurate reconstruction of 3D anatomical surfaces is essential for image-based, patient-specific finite element analysis. Typical problems related to overlapping and intersecting close anatomical structures can be avoided by using semi-automatic deformable models to segment separately different parts of a given structure. However, before generating a finite element grid, it is necessary to merge the resulting triangulations into a single, "water-tight" surface, i.e. with no holes, gaps or self-intersections. This is a problem common to many biomedical engineering disciplines.

The particular case of computational fluid dynamics modeling of blood flow requires an accurate reconstruction of the vessel lumen (Moore et al., 1998,1999). Typically, the finite element grid generation is done either from an analytical representation of the computational domain (Long et al., 2000; Quarteroni et al., 2000; Taylor et al., 1999; Zhao et al., 2000) or directly from a discrete surface representation (Nielsen et al., 1991; Cebral and Löhner, 2001; Ladak et al., 2000). The former approach requires an extra step to fit analytical surface patches after segmentation. The latter approach requires a "water-tight" surface triangulation to define the domain. If objects are defined from overlapping components, it is necessary to generate surface grids over intersecting triangulations prior to the finite element mesh generation. The traditional solution to this non-trivial problem has been to calculate the geometric intersection between the surface triangulations (Lo, 1995; Shostko et al., 1999). The triangulations resulting from this approach tend to have very distorted elements, and cannot account for narrow gaps that should be closed as they are too small for any meaningful fluid simulation.

This paper describes a new algorithm to triangulate intersecting surfaces. It is used in combination with a

*Corresponding author. Tel.: +1-703-993-4078; fax: +1-703-993-4064.

E-mail addresses: jcebral@gmu.edu (J.R. Cebral), lohner@rossini.gmu.edu (R. Löhner), pchoyke@nih.gov (P.L. Choyke), pyim@nih.gov (P.J. Yim).

PII: S0021-9290(01)00018-5

cylindrical deformable model segmentation technique developed by Yim et al. (2001) to create anatomically realistic surface models of arteries from magnetic resonance angiography (MRA) images. These triangulations are then used as support surfaces to generate finite element grids for hemodynamics calculations (Cebral and Löhner, 2001). Adaptive background grids are used to control the accuracy of the procedure. The main advantage of this fully automatic surface-merging algorithm is that it reduces the problem of intersecting triangulations to the extraction of an iso-surface, therefore it is very simple and straightforward to implement. Moreover, it avoids the problem of badly formed triangles and narrow gaps encountered in traditional approaches.

2. Methods

The basis of our merging technique is the representation of the object surfaces as iso-surfaces of a scalar function defined on a background grid. The scalar function is the shortest distance to the object surface. Assigning a negative distance to points that lie inside a surface and a positive distance to those outside, the object surface can be recovered by extracting the iso-surface of zero distance. The merging proceeds by first computing the shortest absolute distance to any object and then extracting the zero-level iso-surface. Any number of objects can be merged simultaneously with this approach.

The algorithm used to generate surface triangulations over intersecting objects can be summarized as follows:

- Read in the surface triangulation of each object.
- Improve the surface triangulations.
- Construct a regular grid of tetrahedra that encloses the objects (voxelization); in the sequel, we will denote this mesh as background grid;
- DO: For each point in the background grid:
- DO: For each object:
- Obtain the shortest distance to the surface triangulation;
- Decide if the point lies inside or outside the object based on the surface normal;
- Assign a negative distance to interior points and a positive distance to exterior points;
- Keep the shortest absolute distance to any object.
- ENDDO
- ENDDO
- Extract the iso-surface of distance equal to zero; this yields the merged or intersected surfaces.
- Improve the final surface triangulation.

The quality of the initial object triangulations may need to be improved in a pre-processing step in cases where the initial surface meshes contain elements of very large aspect ratios. First, *unused points*, i.e. points not touched by any triangle, are deleted. Secondly, *duplicated points* are deleted using an octree data-structure (Löhner, 1988). Two points are considered duplicated if their distance after scaling the whole domain to the unit box is less than a specified tolerance (e.g. 10^{-8}). Then, *degenerated elements*, i.e. elements with a repeated node, are deleted. Degenerated elements may appear after removal of duplicated points. Then, highly distorted or very small elements are removed using an *edge-collapsing* algorithm (Hoppe, 1998). Finally, a *diagonal swapping* scheme is used to minimize the maximum angle of the two triangles adjacent to a given edge (Hoppe, 1998). In both cases, edge collapses or diagonal swappings are prevented if the new configuration changes considerably the normal direction of the surrounding elements. The final triangulation after merging is also post-processed using these optimization schemes.

The initial voxelization of the domain is constructed as a Cartesian grid with cubic voxels. Given the bounding box of the set of objects (with a safety marging) $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$, and a maximum allowable number of points $N_{\max}$ the size of a voxel will be approximately $h = [V/N_{\max}]^{1/3}$, where V is the volume. The final number of points along the x, y, z directions is obtained iteratively by changing the initial number of points in the x, y, z directions obtained from this formula. The points are then joined to form tetrahedra by using stencils. We have used both 1 : 5 and 1 : 6 stencils. Both yield valid results.

The worst-case algorithmic complexity to determine the closest distance from a voxel to a triangle would be $O(N_t x N_p)$ for N_t triangles and N_p background points. In order to reduce the algorithmic complexity to $O(\mathrm{N_p})$, a loop is performed over the triangles. Only the points lying in the vicinity of each triangle are examined. The calculation of the closest distance to the triangulation for each point may be summarized as follows:

- Initialize the minimum distance for points to a large number
- DO: For each object:
- DO: For each triangle:
- Determine the bounding box;
- From the bounding box, determine the region of points to be examined;
- Obtain the points in this region;
- DO: For each of these close points:
- Compute the distance to the triangle
- Assign a negative or positive sign if point is inside or outside the object
- IF smaller absolute distance: store the distance with correct sign
- ENDDO
- ENDDO

- DO: For each point:
- IF new distance smaller than previous: store new distance
- ENDDO
- ENDDO

For a background grid with constant voxel size, the points in the region close to a triangle can easily be determined. For adaptive background grids (see below), octrees are employed. The distance from a point to a given triangle is calculated as explained by Löhner (1996).

The decision of whether a point lies inside or outside a given object is based on the normal to the surface of the object. Once the distance vector $\mathbf{v}$ from a point to the surface is found, the point is considered to be outside if $\mathbf{v} \cdot \mathbf{n} > 0$ and inside if $\mathbf{v} \cdot \mathbf{n} < 0$, where $\mathbf{n}$ is the outer normal of the surface.

Some precautions must be taken in order to avoid misclassification of points. If the triangle "contains" the point (i.e. its shape functions obey $0 \leqslant \xi,\ \eta \leqslant 1$), the triangle normal is taken. If the point falls on an edge of the triangle, the average normal of the two adjacent triangles is taken. And if the point falls on a vertex, the average normal of the triangles surrounding the vertex is taken.

The iso-surface triangulation is obtained from the background grid using the following algorithm. The edges crossed by the iso-value desired (zero) are marked, and the points that will comprise the final triangulation are introduced along them. In a second pass over the elements, the number of edges marked is counted, and either one (three marked edges) or two triangles (four marked points) are introduced. The triangles are oriented so that their normal points in the direction of increasing values of the distance-to-surface function.

The accuracy of the described procedure will depend on the element size of the background mesh. An adaptive background grid, that refines elements close to the surface triangulation, will allow a much better resolution than a uniform voxelization. A procedure first introduced by Löhner and Baum (1992) was implemented in this context. The elements close to the surface triangulation are marked for refinement. These elements are then subdivided further using classic h-refinement. This procedure is extremely fast and the additional accuracy obtained pays off for the increased algorithmic complexity. Given that the points of the new surface lie exactly on the original surfaces, the maximum distance between the two triangulations will be a fraction of the background grid element size.

Once individual objects have been merged into a single, "water-tight" surface triangulation, finite element grids can be generated using this surface to define the computational domain, i.e. as a support surface. Depending on the needs of the specific applications, different element types (tetrahedra, hexahedra, triangles, quadrilaterals, etc.) and element size distributions can be specified for the final finite element grid. For hemodynamics calculations, volumetric grids of tetrahedra can be generated directly from surface triangulations using the algorithms described by Cebral and Löhner (2001) and Löhner (1996).

3. Results

The methodology was first applied to the merging of surface models representing the internal and external carotid arteries of a normal subject. Each branch was individually reconstructed from contrast-enhanced MRA images of a normal volunteer using a cylindrical deformable model (Figs. 1a and b). A fine background

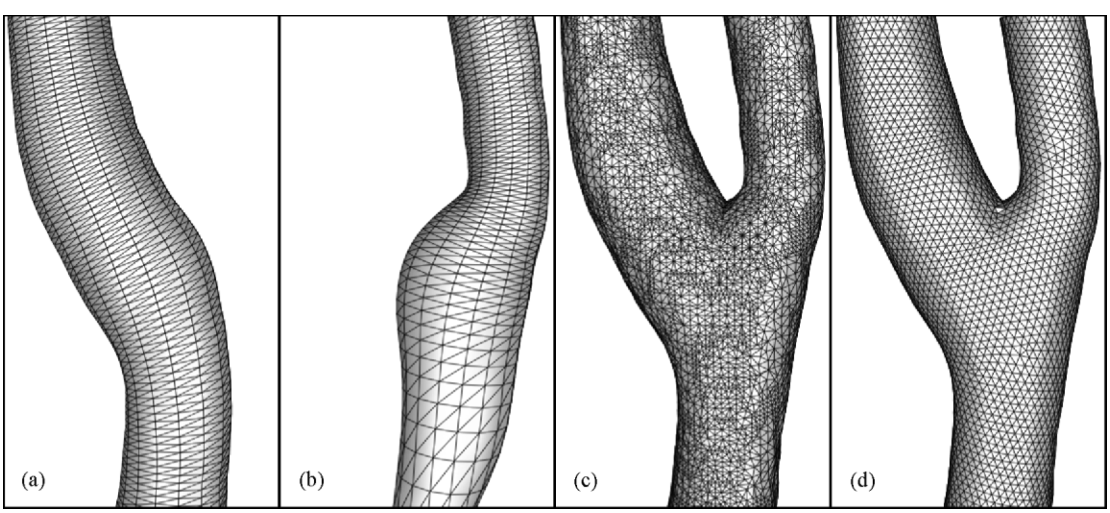

Fig. 1. Normal subject: (a) reconstructed internal carotid branch; (b) reconstructed external carotid branch; (c) carotid bifurcation model after merging; (d) surface of the finite element mesh.

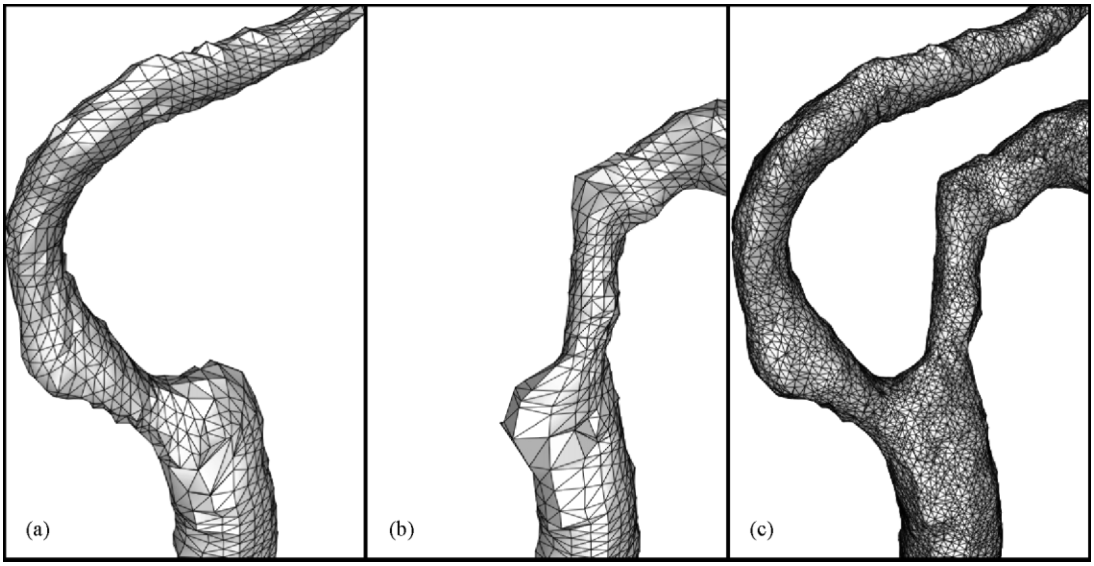

Fig. 2. Patient with stenosis: (a) reconstructed internal carotid branch; (b) reconstructed external carotid branch; (c) surface triangulation of the merged internal and external branches of the carotid artery.

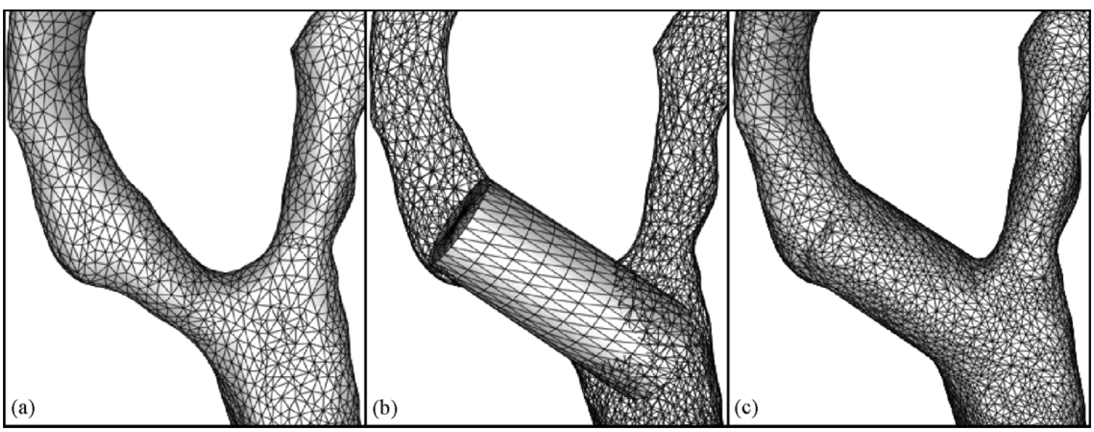

Fig. 3. (a) Carotid artery with stenosis before stenting; (b) location and geometry of idealized vascular stent; (c) carotid artery model after introduction of virtual stent.

grid with no adaptation was used to generate a surface triangulation over the intersecting arterial branches (Fig. 1c). This merged model was then used to generate a volumetric finite element grid composed of tetrahedral elements with a uniform element size distribution (Fig. 1d).

Secondly, the surface merging algorithms were used to generate a surface model of the carotid artery bifurcation of a patient with carotid artery stenosis. Again, both the internal and external branches of the right carotid artery were separately reconstructed from contrast-enhanced MRA images using deformable models (Figs. 2a and b). Since the original triangulations had poor element quality, they were optimized before merging. An adaptive background grid with two levels of refinement was used to merge these surface triangulations. The surface of the merged model contained about 3.2 K points (Fig. 2c).

Finally, the usefulness of the merging technique to modify vessel geometries in order to simulate interventional procedures was demonstrated in the case of the introduction of a vascular stent into the stenosed internal carotid artery. The geometry of the stenosed carotid artery bifurcation (Fig. 3a) was modified by adding (merging) an idealized vascular graft into the internal carotid branch (Fig. 3b). The final merged model (Fig. 3c) can then be used to generate finite element grids for conducting fluid dynamics calculations in order to predict the outcome of the stenting procedure. This type of simulations may potentially be used to evaluate the efficacy of different therapeutic alternatives.

4. Discussion

The combination of the present surface-merging algorithm with segmentation procedures based on deformable models can be successfully used to construct anatomically realistic models free of intersections with other close structures. These surface triangulations, which are free of badly formed triangles, can then be used to generate 3D grids for finite element analysis.

Since the surface-merging approach is based on the extraction of the iso-surface of zero distance, the initial object triangulations must be closed in order to classify points as interior or exterior to the surface and assign the correct sign. Moreover, this classification is done by looking at the surface normal, therefore all objects must be oriented in the same way, the surface triangulations must not contain any duplicated points, degenerated elements or very small elements which may lead to an erroneous normal. The accuracy of the procedure is limited by the resolution of the background grid, and can be controlled by adaptive refinement of elements close to the surfaces. It must also be noted that the algorithms were designed for smooth surfaces and that surface features such as ridges or corner points present in the initial triangulations are not necessarily preserved in the merged surface.

The methodology was demonstrated in the case of triangulating intersecting branches of carotid arteries individually reconstructed from MRA images of human subjects. Since the merged surface is constructed by extracting an iso-surface of zero distance to the object surfaces, the points in the new triangulation lie exactly on the original surface. Therefore, the maximum distance between the two triangulations is a fraction of the background grid element size. The accuracy of the method is controlled by adaptively refining background grid elements close to the object surfaces. Thus, the background grid can be automatically adapted to ensure that the maximum distance between the two surfaces is less than a given user-defined tolerance. The merged surfaces satisfy the requirements for finite element grid generation, namely they are "water-tight", i.e. contain no holes, gaps, overlapping triangles or self-intersections. It was also demonstrated that the present methodology can handle, with no extra difficulty, cases that require the introduction of extra geometrical features or modification of the vessel geometry and topology such as the simulation of vascular stenting, balloon angioplasty, bypass grafts, etc.

Although the merging technique was applied to blood vessels, it is general and can be used in other areas.

References

Cebral, J.R., Lohner, R., 2001. From medical images to anatomically accurate finite element grids. International Journal for Numerical Methods in Engineering, in press.

Hoppe, H., 1998. Efficient Implementation of progressive meshes. Computers & Graphics 22 (1), 27–36.

Ladak, H.M., Milner, J.S., Steinman, D.A., 2000. Rapid 3D segmentation of the carotid bifurcation from serial MR images. Journal of Biomechanical Engineering 122 (1), 96–99.

Lo, H.S., 1995. Automatic mesh generation over intersecting surfaces. International Journal for Numerical Methods in Engineering 38, 943–954.

Löhner, R., 1988. Some useful data structures for the generation of unstructured grids. Communications in Applied Numerical Methods 4, 123–135.

Löhner, R., 1996. Regridding Surface Triangulations. Journal of Computational Physics 126, 1–10.

Löhner, R., Baum, J.D., 1992. Adaptive H-refinement on 3-D unstructured grids for transient problems. International Journal for Numerical Methods in Fluids 14, 1407–1419.

Long, Q., Xu, X.Y, Bourne, M., Griffith, T.M., 2000. Numerical study of blood flow in an anatomically realistic aorto-iliac bifurcation generated from MRI data. Magnetic Resonance in Medicine 43, 565–576.

Moore, J.A., Steinman, D.A., Ethier, C.R., 1998. Computational blood Flow modeling: errors associated with reconstructing finite element models from magnetic resonance images. Journal of Biomechanics 31, 179–184.

Moore, J.A., Steinman, D.A., Holdsworth, D.W., 1999. Accuracy of computational hemodynamics in complex arterial geometries reconstructed from magnetic resonance imaging. Annals of Biomedical Engneering 27, 32–41.

Nielsen, P.M.F., Le Grice, I.J., Smail, B.H., Hunter, P.J., 1991. Mathematical model of geometry and fibrous structure of the heart. American Journal of Physics 260, H1356–H1378.

Quarteroni, A., Tuveri, M., Veneziani, A., 2000. Computational vascular fluid dynamics: problems, models and methods. Computing and Visualization in Science 2, 163–197.

Shostko, A., Löhner, R., Sandberg, W.C., 1999. Surface triangulation over intersecting geometries. International Journal for Numerical Methods in Engineering 44, 1359–1379.

Taylor, C.A., Draney, M.T., Ku, J.P., Parker, D., Steele, B.N., Wang, K., Zarins, C.K., 1999. Predictive medicine: computational techniques in therapeutic decision-making. Computer Assisted Surgery 4, 231–247.

Yim, P.J., Cebral, J.R., Mullick, R., Choyke, P.J., 2001. Vessel Surface Reconstruction with a tubular deformable model. IEEE Transaction Medical Imaging, submited for publication.

Zhao, S.Z., Xu, X.Y., Hughes, A.D., Thom, S.A., Stanton, A.V., Ari, B., Long, Q., 2000. Blood flow and vessel mechanics in a physiologically realistic model of a human carotid arterial bifurcation. Journal of Biomechanics 33, 975–984.