Hole Patching in Unstructured Mesh Using Volumetric Diffusion

Amitesh Kumar[1] and Alan M. Shih[2]

[1] University of Alabama at Birmingham, AL, USA amitesh@uab.edu
[2] University of Alabama at Birmingham, AL, USA ashih@uab.edu

Summary. A three-dimensional object of interest in an engineering analysis, using mesh-based computational technology, requires watertight surface geometry. However, obtaining watertight surfaces is challenging due to presence of deficiencies or artifacts such as gaps, holes etc. on the surfaces. This research note presents implementation of a volumetric approach for repairing those surfaces which otherwise could not simply be repaired using existing surface-based techniques due to the geometric and topological complexities of holes. The geometry repair approach is based on diffusion equation solved using CFD techniques based on explicit forward difference scheme in time and central difference scheme in space. An algebraic interpolation approach using coarse grid to initialize the flow field for the fine grid has been implemented to speed up solution convergence. Additionally the solver has been multi-threaded using OpenMP library to further speed up the solution process. The result of the solution process and the performance gain using multi-threading has also been demonstrated.

Keywords: geometry repair, volume based approach, volumetric diffusion, OpenMP.

1 Introduction

Preprocessing steps in simulations using mesh-based computational technologies, such as Computational Fluid Dynamics (CFD) and Computational Structural Mechanics (CSM), involve geometry preparation and mesh generation. However, it has been a challenging task to take a geometry and turn it into a high quality mesh. This is because there could be deficiencies in geometry and the configuration of the geometry can make the entire process difficult. Such deficiencies can be in the forms of gaps, holes, protrusions or intersection and overlaps. Repairing deficiencies in the geometry can be a tedious and laborious process. Therefore, how to obtain a watertight geometry ready for mesh generation is an important issue in computational engineering.

Regardless of the source of the geometry data, or the form in which they are represented (parametrically or discretely), the geometries obtained can have many defects due to the data conversion errors or ambiguities in the process.

The most common type of mesh defects or artifacts encountered are holes or isles, singular vertex, handle, gaps and small overlaps, large overlaps, inconsistent orientation, complex edges and intersections [1]. Some of these artifacts like complex edges have a precise meaning while the distinction between small scale and large scale overlaps are described intuitively rather than by strict definition. A number of research papers have tried different approaches in an attempt to address this issue in a more automated and intelligent manner. These approaches broadly fall in two main categories: volume-based repair methods and surface-based repair methods.

The key to all volume based methods lies in converting a surface model into an intermediate volumetric representation from which the output model is then extracted. A flag at each voxel of the volumetric representation is generated representing whether the particular voxel lies inside, outside or on the surface of the geometry. The interface between inside and outside cells then define the topology and geometry of the reconstructed model. Due to their very nature, volumetric representations do not allow for artifacts like intersections, holes, gaps or overlaps or inconsistent normal orientation [1]. Volumetric algorithms are typically fully automatic and produce watertight models and depending on the type of volume, they can often be implemented very robustly.

Due to their very nature, volumetric representations do not allow for artifacts like intersections, holes, gaps or overlaps or inconsistent normal orientation. Volumetric algorithms are typically fully automatic and produce watertight models and depending on the type of volume, they can often be implemented very robustly. As a result in some cases it becomes necessary to use volume based approach to repair model. In this work we are working on developing a robust and fully automatic volume based repair method which could handle dirty geometry with holes, isles and intersections which is based on numerical solution of Diffusion Equation in a 3D volume grid.

2 Diffusion Equation

Diffusion is a time-dependent process, constituted by random motion of given entities and causing the statistical distribution of these entities to spread in space. The concept of diffusion is tied to notion of mass transfer, driven by a concentration gradient. The diffusion equation can be obtained easily from Continuity equation when combined with the Fick's first law. Diffusion equation is a partial differential equation continious in both space and time and can be simplified and written as,

$$\frac{\partial \phi}{\partial t} = \alpha \Delta^2 \phi \tag{1}$$

Where α is a constant. Using *forward difference* scheme in time and *central difference* scheme in space based on Taylor's series expansion, the diffusion equation for a Cartesian grid simplifies to,

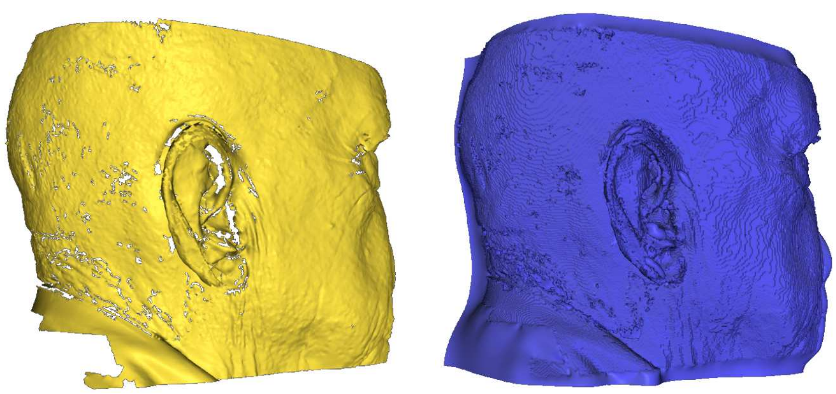
$$\phi^{n+1}_{i,j,k} = \phi^{n}_{i,j,k} + \alpha \Delta t \begin{bmatrix} \left(\frac{\phi^{n}_{i+1,j,k} - \phi^{n}_{i,j,k}}{\Delta x^2} - \frac{\phi^{n}_{i,j,k} - \phi^{n}_{i-1,j,k}}{\Delta x^2} \right) \\ + \left(\frac{\phi^{n}_{i,j+1,k} - \phi^{n}_{i,j,k}}{\Delta y^2} - \frac{\phi^{n}_{i,j,k} - \phi^{n}_{i,j-1,k}}{\Delta y^2} \right) \\ + \left(\frac{\phi^{n}_{i,j,k+1} - \phi^{n}_{i,j,k}}{\Delta z^2} - \frac{\phi^{n}_{i,j,k} - \phi^{n}_{i,j,k-1}}{\Delta z^2} \right) \end{bmatrix} \quad (2)$$

where $\Delta x, \Delta y$ and Δz are the grid spacings of a Cartesian grid in x, y and z directions respectively. Equation (2) provides the explicit numerical solution which is central difference in space and forward difference in the time domain for the diffusion equation given in equation (1).

After performing stability analysis on our numerical scheme using Von Neumanns analysis and Courant–Friedrichs–Lewy (CFL) condition for the stability of the numerical scheme in three dimensions (3D), we come with the inequality defining the range of values for the constants given in equation (2).

$$0 \leq \alpha \Delta t < \frac{\Delta x^2_{min}}{6} \quad (3)$$

We define the Cartesian grid with the analogy of a curved thin plate in 3D space whose one side is heated while the other side is cold. Diffusion equations presented earlier are used to find a steady state solution.



(a) surface with numerous holes and complex topology

(b) Watertight surface after running the solver and extracting surface mesh

Fig. 1. Front view of the *Ear Data*

3 Result

The diffusion solver was run on a linux machine against a number of non-manifold models. One of these is what we call as the "*Ear Data*". This surface model was obtained by scanning the face of a patient at UAB hospital. The scan came with a number of discontinuities, islands and overlaps, which left much to be desired. Figure 1(a) , in the left, shows rendering of the original surface. Figure 1(b) shows the surfaces obtained from Cartesian grid after running the solver. The coarse and fine Cartesian grids had resolutions of 72x90x100 and 285x360x400 voxels respectively. In this problem, the solver was run for 3000 iteration on a coarse Cartesian grid and 200 iterations on the finer Cartesian grid. The result of the diffusion solver from the coarse Cartesian grid was interpolated on the finer Cartesian grid as the initial value for the solver to obtain a faster convergence of the numerical solution.

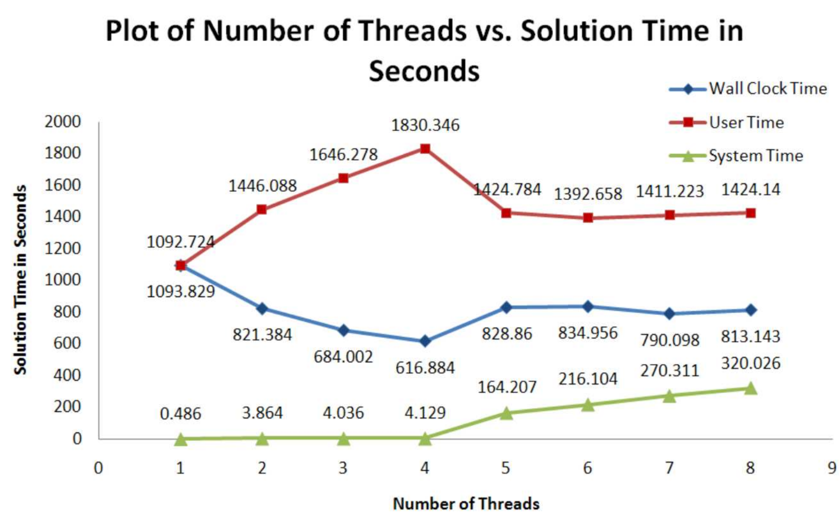


Fig. 2. Plot of number of threads vs. Ssolution time using OpenMP library for multi-threading

This approach has inherent parallelism as we are using Carteisan grids. Our code was run both in single threaded multi threaded configuration using OpenMP library for assessing performance gain. Figure 2 shows the plot of number of threads vs. Solution time for the solver. What we are interested in the wall-clock time required for the solver to run. We notice that with the number of the threads as four, the solution time is minimum for the particular example. The repaired surface is extracted as a post processing step after numerical solution has been obtained. Marching Cubes algorithm [2] has been used for surface extraction from Cartesian grid. It could however be noticed that Marching Cubes introduces aliasing artifacts (*staircasing*) on the surface as the algorithm is feature insensitive.

4 Conclusion

We have presented a volume based approach which tries to repair a discrete geometry by solving diffusion equations. The work is still in progress. However for proof of concept we have voxelized the whole discrete geometry into one single Cartesian grid while running the solver. We have shown the result of our method using a model with complex topology.

Volume based approaches like the one described in this research can be used to repair the models with artifacts that surface based models otherwise cannot robustly handle. They however also pose some potential problems. The conversion to and from a volume leads to resampling of the model. It however often introduces aliasing artifacts, loss of model features and destroys any structure that might have been present in the connectivity of the input model. The focus of this work is to address the need to develop a method to obtain a watertight geometry from a geometric model that has the presence of holes of complex topology. Despite all their shortcomings volumetric algorithms can solve some problems of complex topology robustly in geometry repair which is not possible in surface based approaches.

References

1. Botsch M., Pauly M., Kobbelt L., Alliez P., Lévy B., Bischoff S. and Rössl C.: Geometric Modeling Based on Polygonal Meshes. ACM SIGGRAPH 2007 Courses - International Conference on Computer Graphics and Interactive Techniques, (2007).
2. Lorenson E. W., Cline E. H.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. ACM Computer Graphics, 21(4), (1987).
3. Nooruddin F.S., Turk G.: Simplification and repair of polygonal models using volumetric techniques. IEEE Transactions on Visualization and Computer Graphics, 9(2):191-205 (2003).
4. Bischoff S., Pavic D., and Kobbelt L.: Automatic restoration of polygon models. Transactions on Graphics, 24(4):1332-1352 (2005).
5. Murali M. T. and Funkhouser A. T.: Consistent solid and boundary representations from arbitrary polygonal data. In Proc. Symposium on Interactive 3D Graphics, 155-162 (1997).
6. Davis, J., Marschner, S. R., Garr, M. and Levoy, M.: Filling Holes in Complex Surfaces using Volumetric Diffusion. Proceedings of First International Symposium on 3D Data Processing, Visualization, Transmission, 428-861 (2002).
7. Curless, B. and Levoy M.: A Volumetric Method for Building Complex Models from Range Images. Computer Graphics, 30:303-312 (1996).
8. Ju, T.: Robust Repair of Polygonal Models. Proceedings of ACM SIGGRAPH, 2004 ACM Transactions on Graphics, 23:888-895 (2004).