2.2

A Hole-filling Algorithm Using Non-uniform Rational B-splines

Amitesh Kumar, Alan Shih, Yasushi Ito, Douglas Ross and Bharat Soni

Department of Mechanical Engineering
University of Alabama at Birmingham, Birmingham, AL, U.S.A.
{amitesh, ashih, yito, dhross, bsoni}@uab.edu

Abstract

A three-dimensional (3D) geometric model obtained from a 3D device or other approaches is not necessarily watertight due to the presence of geometric deficiencies. These inadequacies must be repaired to create a valid surface mesh on the model as a pre-process of computational engineering analyses. This procedure has been a tedious and labor-intensive step, as there are many kinds of deficiencies that can make the geometry to be non-watertight, such as gaps and holes. It is still challenging to repair discrete surface models based on available geometric information. The focus of this paper is to develop a new automated method for patching holes on the surface models in order to achieve watertightness. It describes a numerical algorithm utilizing Non-Uniform Rational B-Splines (NURBS) surfaces to generate smooth triangulated surface patches for topologically simple holes on discrete surface models. The Delaunay criterion for point insertion and edge swapping is used in this algorithm to improve the outcome. Surface patches are generated based on existing points surrounding the holes without altering them. The watertight geometry produced can be used in a wide range of engineering applications in the field of computational engineering simulation studies.

Key Words: Hole Filling, NURBS, Geometry Repair, Watertight, Mesh Generation, CFD, CSM, Delaunay Triangulation, Edge Swapping

1. Introduction

Mesh-based computational technologies, such as Computational Fluid Dynamics (CFD) and Computational Structural Mechanics (CSM), require a high quality mesh to achieve numerical stability and accuracy. Most of mesh generators need a properly prepared underlying geometry so that a valid surface mesh respecting the geometry definition can be generated. The geometry needs to be watertight. Overlapping or gaps between geometry entities, for example, must be treated properly before the mesh generation process. A geometry or surface mesh is considered to be not watertight in two situations:

1. It has edges shared by only one polygon, *i.e.*, they lie on boundaries. The occurrences of this kind of connected edges create holes in the surface.
2. It has edges shared by more than two polygons. The occurrences of these kinds of edges create non-manifold by virtue of hanging triangles. This kind of non-manifold mostly occurs in Computer-Aided Design (CAD) applications due to improper stitching of surface patches to generate a desired geometric model.

The surface mesh not only needs to maintain its mesh quality by achieving the required geometric quality measures, but it also needs to represent the geometry with high fidelity. It is still challenging to repair defective geometric surface models automatically and robustly while maintaining high geometric fidelity. This is especially true for the holes and gaps found on discrete surface models, as we have limited information upon which to estimate the missing geometric information.

Several algorithms have been proposed for filling polygonal holes [1-11]. Most of these algorithms cater to very specific sets of problems, *i.e.*, patching holes from only a particular source of geometry [1-6]. These algorithms broadly fall in two main categories: volume-based repair methods [5-7] and mesh-based repair methods [1-4, 8-11].

The key to all volume based methods lies in converting a surface model into a volume representation and generating a sign at each voxel representing whether the particular voxel lies inside, outside or on the surface of the geometry. The signs are generated with the help of distance map of each point on the geometry using line-of-sight information, which is usually obtained from range-finding devices. This crucial piece of information may not be available for a purely computational geometric model. The uncertain voxels are assigned signs based on volumetric diffusion [5, 6]. Once all the voxels are assigned signs, the volume-based methods simply extract the contour to find a closed surface. Curless and Levoy [6] propose a hole

filling algorithm based on volumetric diffusion optimized for patching holes in models reconstructed by range-finding devices. These devices generally create small holes with complex topologies. However, these holes can be large compared to of the size of the polygons in the mesh. The volumetric methods are not suitable for this research because of the lack of line-of-sight information. Ju [7] presents a method for generating signs of voxels for repairing a polygonal mesh using an octree. However, this method is suitable only for small sized holes.

A number of algorithms have been suggested for filling holes in a triangular mesh using a surface triangulation-based approach. Leipa [8] describes a method for filling holes by a weight-based hole triangulation, mesh refinement based on the Delaunay criterion and mesh fairing based on energy minimization as used in [9]. The algorithm was confined to holes in an oriented connected mesh in relatively smooth region with the assumption that the holes are relatively small compared to the entire model. Barequet and Sharir [1] use a dynamic programming method to find minimum area of triangulation of a three-dimensional (3D) polygon in order to fill mesh holes. Barequet and Kumar [2] describe an interactive system that closes small cracks by stitching corresponding edges and fills big holes by triangulating the hole boundary similar in approach to Barequet and Sharir [1]. Unfortunately, this method cannot provide satisfactory results on relatively large-sized holes with complex geometric shapes. Jun [4] describes an algorithm based on stitching planar projection of complex holes and projecting back the stitched patch. Bruno [10] attempts to fill a hole and blend surface based on global parameterization for complete geometry approximation and then energy minimization for surface blending based on the assumption that global parameterization of the complete model is available or possible. Branch *et al.* [11] suggest a method for filling holes in triangular meshes using a local radial basis function. The method works quite well with skinny holes, but fails miserably when the holes are fatter in shape. All these algorithms are limited by their assumption that holes are relatively small-sized.

In this paper, a new automated method is proposed for filling holes on a triangulated surface model to make it watertight [12]. The existing points around the holes are used to obtain a set of Non-Uniform Rational B-Splines (NURBS) surfaces approximating the missing smooth surface patches. A Delaunay triangulation method is used to generate internal points which are then projected on to the set of NURBS surfaces to obtain the desired patch. The patches generated by this method are achieved without altering the geometric information of the surrounding geometry. This algorithm is currently applicable to topologically simple holes in the discrete geometry as a triangular mesh, while holes with more sophisti-

cated topology will be considered in the future study. Such topologically simple holes are common in the geometry obtained from 3D scanners or geometry extraction from the medical image datasets using the marching cubes algorithm [14].

2. Hole Filling Processes

In order to properly fill the holes presented in the geometry, we need to first identify the boundaries of these holes, obtain its neighboring geometric information and attempt to estimate the missing surface patches. For the description purpose, a generic surface with a circular hole as shown in Figure 1 is used as a test case.

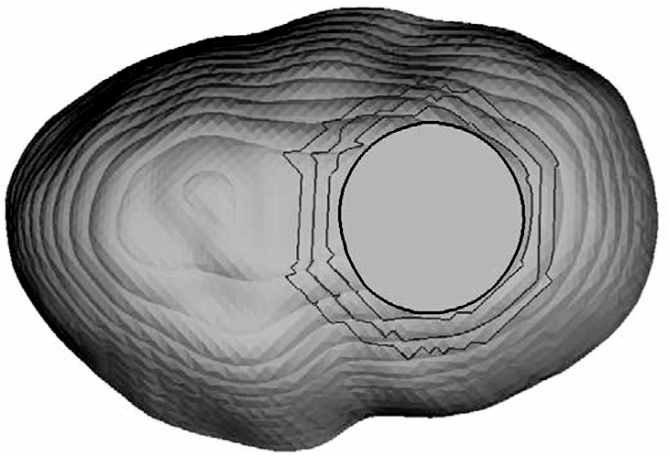


Figure 1. Discrete ordered points in the form of rings

2.1. Identification of a Hole

A hole is identified by checking the edge list to locate each boundary edge that is shared by only one triangle. A set of the boundary edges creates a hole on the surface. Once a closed edge list is identified, these edges are used to perform initial triangulation using the vtkTriangleFilter algorithm in the Visualization ToolKit [15]. Point insertion and edge swapping based on the Delaunay criterion are repeated on the initially triangulated mesh to produce a fine mesh that approximately (but not accurately) represents the

missing hole surface. The fine mesh will be used later in Section 2.4 to project onto a more accurately approximated surface.

2.2. Identification of Neighboring Geometry into Ring Curves

The points on the original discrete surface are selected to form rings around the hole identified in Section 2.1. The hole is considered as 0th ring and only the 2nd, 4th and 6th rings from the hole are used to avoid the NURBS representation of the rings crossing each other. The rings are represented by NURBS curves to form a set of NURBS surfaces approximating the missing surface in the hole region (see Section 2.3). Figure 1 shows part of a surface with a hole in black and the rings in blue, red and green around it. The approach is robust even when two holes lie in the vicinity of each other as the rings would simply go around the boundary of the other hole.

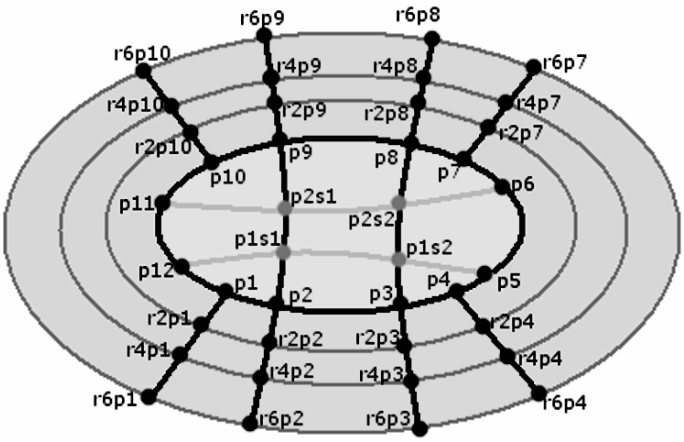


Figure 2. A simplistic NURBS representation of a hole, rings surrounding it and various points lying on them

2.3. Non-Uniform Rational B-Spline (NURBS) Surfaces

Once the rings around the hole are obtained as described in Section 2.2, they are used to construct a set of NURBS surfaces approximating the missing region. This is first done by calculating spline curves of the rings to ensure that they all possess the same number of points, hence producing

smoother curves. Each of these closed curves is then divided into four segments at u values of 0.0, 0.25, 0.5 and 0.75 to form a four-sided surface inside, as illustrated in Figure 2, where u is the parametric value used in the NURBS curve formulation [13].

Let the NURBS curve representation of the hole and the 2nd, 4th and 6th rings be represented by R_0, R_2, R_4 and R_6, respectively. Let P be the set of 12 points found on R_0 by using a set of scalar values u_i (i = 1, 2,…, 12) in the parametric NURBS curve. A parameter, t, increases with each incremental rotation of the rings around the hole.

$$P = \{p_1, p_2, \ldots, p_{12}\}$$
$$u_i = \begin{cases} u_i' & (0 \le u_i' \le 1) \\ u_i' - 1 & (u_i' > 1) \end{cases}, \quad u_i' = t + \frac{i-1}{12} \quad (0 \le t \le 1) \tag{1}$$

Each of the points p_2, p_3, p_8 and p_9 is projected on the NURBS curve approximation of rings R_2, R_4 and R_6 using nearest point projection to obtain three sets of four additional points designated as $\{r_2p_2, r_2p_3, r_2p_8, r_2p_9\}$, $\{r_4p_2, r_4p_3, r_4p_8, r_4p_9\}$ and $\{r_6p_2, r_6p_3, r_6p_8, r_6p_9\}$, respectively. These points are used to obtain two ordered sets of control points S_1 and S_2:

$$\begin{aligned} S_1 &= \{r_6p_2, r_4p_2, r_2p_2, p_2, p_9, r_2p_9, r_4p_9, r_6p_9\} \\ S_2 &= \{r_6p_3, r_4p_3, r_2p_3, p_3, p_8, r_2p_8, r_4p_8, r_6p_8\} \end{aligned} \tag{2}$$

S_1 and S_2 are used to create two NURBS curves, N_{S1} and N_{S2}, and they are used to obtain four additional points in the hole region designated as p_1s_1, p_2s_1, p_1s_2 and p_2s_2 by varying u values of the NURBS curves N_{S1} and N_{S2}. Let u_2 and u_9 be the parametric positions of points p_2 and p_9, respectively, on NURBS curve N_{S1}. Parametric positions of points p_1s_1 and p_2s_1 are obtained as $u_2 + \frac{1}{3}(u_9 - u_2)$ and $u_2 + \frac{2}{3}(u_9 - u_2)$, respectively. Points p_1s_2 and p_2s_2 on N_{S2} are obtained in the same way.

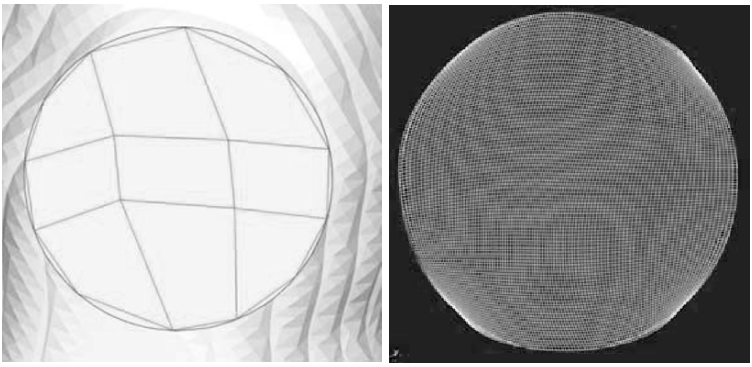
Let S_3, S_4, S_5 and S_6 be four sets of control points given as follows:

$$\begin{aligned} S_3 &= \{p_1, p_{12}, p_{11}, p_{10}\} \\ S_4 &= \{p_2, p_1s_1, p_2s_1, p_9\} \\ S_5 &= \{p_3, p_1s_2, p_2s_2, p_8\} \\ S_6 &= \{p_4, p_5, p_6, p_7\} \end{aligned} \tag{3}$$

These four sets of control points are used to obtain four interpolating NURBS curves, which are collectively used to create a lofted NURBS surface as shown in Figure 3.

A number of lofted NURBS surfaces are obtained by varying t. In this paper, 18 control surfaces were created for each hole in an attempt to ob-

tain better estimation for the missing region. When each of the ring curves is divided into four segments for constructing a NURBS surface, the starting points are obtained by varying t corresponding to 10° of increment (t = 0, 1/18, 2/18,…, 17/18). It can be further studied to improve the algorithm efficiency by reducing the number of NURBS surfaces needed while maintaining the quality of the final surface generated.



(a) (b)

Figure 3. NURBS surface and control net: (a) Control net in the hole region; (b) Lofted NURBS surface and underlying control surface

2.4. Projection of Interior Points

All interior points generated in Section 2.1 are projected on to a set of 18 lofted NURBS surfaces to find a set of 18 coordinates. A simple average of these 18 projections is used to find the final coordinates and form a good approximation for the hole region. Figure 4 shows patches generated using only one projection and eighteen projections respectively. It clearly shows a distinct difference in the patches where the one generated with only a single NURBS surface tends to be very sensitive to the manner in which the ring curves are split, while the averaged one reflects the geometric information from the neighboring geometry more accurately. This is clear when the patch is placed on the original surface, as shown in Figure 5: the ridge in the middle of Figure 4b reflects the ridge on the neighboring geometry. The connectivity of each point on the mesh remains the same as the original point after the projection.

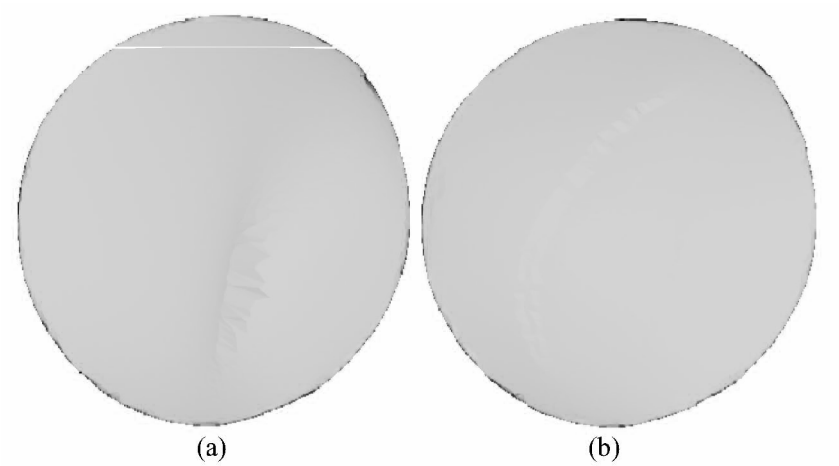


Figure 4. Patches generated using (a) one projection and (b) eighteen projections

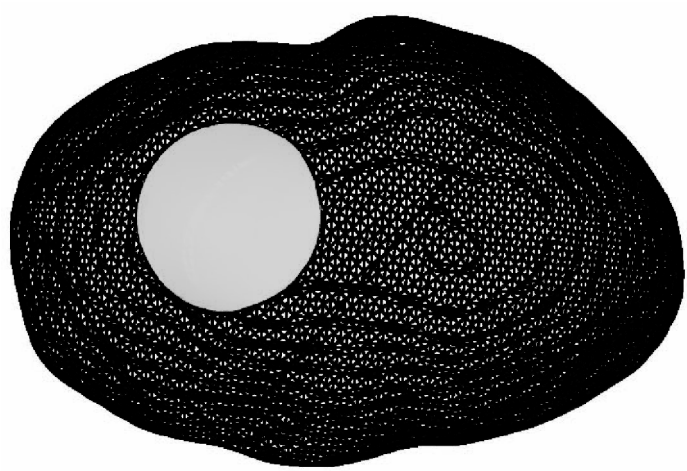


Figure 5. Surface patch generated for the hole. It reflects the ridge line feature that is present in the neighboring geometry.

3. Benchmark

In order to evaluate the errors of this algorithm, ellipsoids are chosen as benchmark cases to calculate the average errors in terms of radius and the

standard deviation of errors. An ellipsoid of semi-axes *a*, *b* and *c* is defined as:

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1 \tag{4}$$

Error in the location of points on the ellipsoids can be quantified as

$$E_i=\frac{x_i^{\,2}}{a^2}+\frac{y_i^{\,2}}{b^2}+\frac{z_i^{\,2}}{c^2}-1 \tag{5}$$

The average error can be measured as:

$$\overline{E}=\frac{\sum_i^N E_i}{N} \tag{6}$$

The standard deviation is the measure of the spread in a set of values. The standard deviation in the error of coordinate position on the patches can be obtained as follows:

$$\sigma=\sqrt{\frac{\sum_i^N\left(\overline{E}-E_i\right)^2}{N}} \tag{7}$$

Figure 6. Patches on a sphere with *a* = *b* = *c* =1

Figure 7. Patches on an ellipsoid with *a* = 1, *b* = 0.5 and *c* =1

Figure 8. Patches on an ellipsoid with *a* = 1, *b* = 0.2 and *c* =1

A sphere is a special case of an ellipsoid when $a = b = c$. The sphere with a radius of 1 is shown in Figure 6. Figures 7 and 8 show two ellipsoids obtained by changing b values. Each of these three ellipsoids has 26,826 nodes and four holes that have been filled by patches (green) using the proposed algorithm. Table 1 shows average errors and standard deviations in the position of points on the patches generated for the three cases. It shows that the average errors lie within 0.35%, while the standard deviations are within 0.5%. We can also notice that the standard deviation increases with the increasing curvature of the surfaces surrounding the hole. This can be seen by comparing Figure 6, which has the lowest curvature, with Figure 8, which has the highest curvature.

Table 1. Average errors and standard deviations in the position of points on the patches

Cases	# of new nodes	$\overline{E}$	σ
$a = b = c = 1$ (sphere)	8,500	-3.16299×10^{-3}	2.71724×10^{-3}
$a = 1$, $b = 0.5$, $c = 1$	8,607	-2.91879×10^{-3}	3.52282×10^{-3}
$a = 1$, $b = 0.2$, $c = 1$	11,838	-2.97327×10^{-3}	4.84794×10^{-3}

4. Applications

This section aims to demonstrate the proposed algorithm with several complex geometric models. Section 4.1 shows a model of white matter of a human brain with 13 holes of various sizes punched on its surface. Section 4.2 shows a human pelvis mesh with 10 holes. These models were chosen to demonstrate the robustness of the algorithm in patching holes in areas of high curvature as well as larger-sized holes in the manner that the patches reflect the neighboring surface characteristics.

4.1. Brain White Matter Model

Figure 9 shows a simplistic model of white matter of a human brain with holes on its surface [16]. The model was obtained after segmentation of a series of Magnetic Resonance Imaging (MRI) data [17]. Once the segmentation was done, the marching cubes algorithm was used to obtain the triangulated surface. Thirteen holes were punched on the white matter model—some of them in regions of very high curvature—to test the robustness of the algorithm. The proposed algorithm successfully generated smooth patches to fill all the holes. The resulting surface is shown in

Figure 10. This test case emphasized the capability of this algorithm to generate smooth patches for holes with challenging geometry and curvature without smoothing out the local features of the geometry.

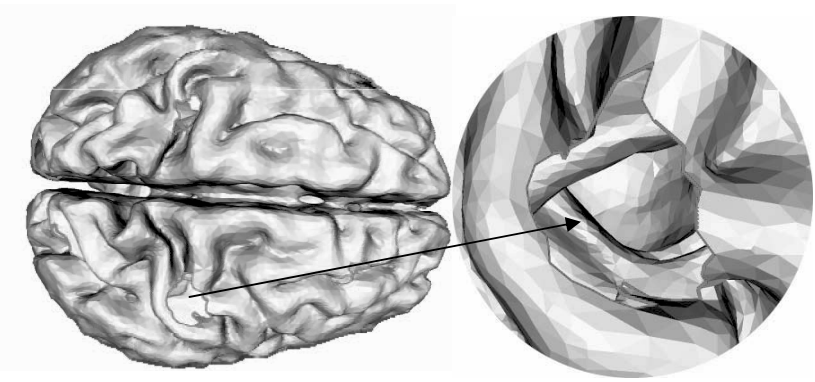


Figure 9. White matter model with 13 holes on its surface

4.2. Pelvis

Figure 11 shows a surface rendering of a patched unstructured surface mesh of a human pelvis [16]. The pelvis model is made of 1,090,700 triangles and 546,093 nodes with ten holes. The patches for the ten holes are made of 31,099 triangles and 16,331 nodes in total. Upon looking closely, it can be noticed that the three patches in dark green, red and yellow provide patches conforming to the surrounding geometry. Although they must be flat surfaces, the proposed method does not have a capability to identify such surfaces. A graphical user interface (GUI) will be added to exclude these holes beforehand.

5. Conclusion and Future Work

A new fully-automated hole-filling algorithm was proposed for triangulated surface models and was applied to several cases successfully. It utilizes the neighboring geometric information by identifying rings of curves around the holes to create a set of NURBS surfaces, which are subsequently used for projection of the triangulated surface patches. The 18 sets of projected coordinates are then averaged to obtain the final point coordinates for the hole region. Several benchmark cases are studied to assess the

accuracy of the algorithms with satisfactory results. Finally, complex biomedical geometries are used to test the robustness of the algorithms. Further investigation will improve the algorithm in terms of computation efficiency and applicability to topologically complex holes.

Acknowledgments

This research is supported in part by the NASA Constellation University Institutes Project (CUIP) No. NCC3-994.

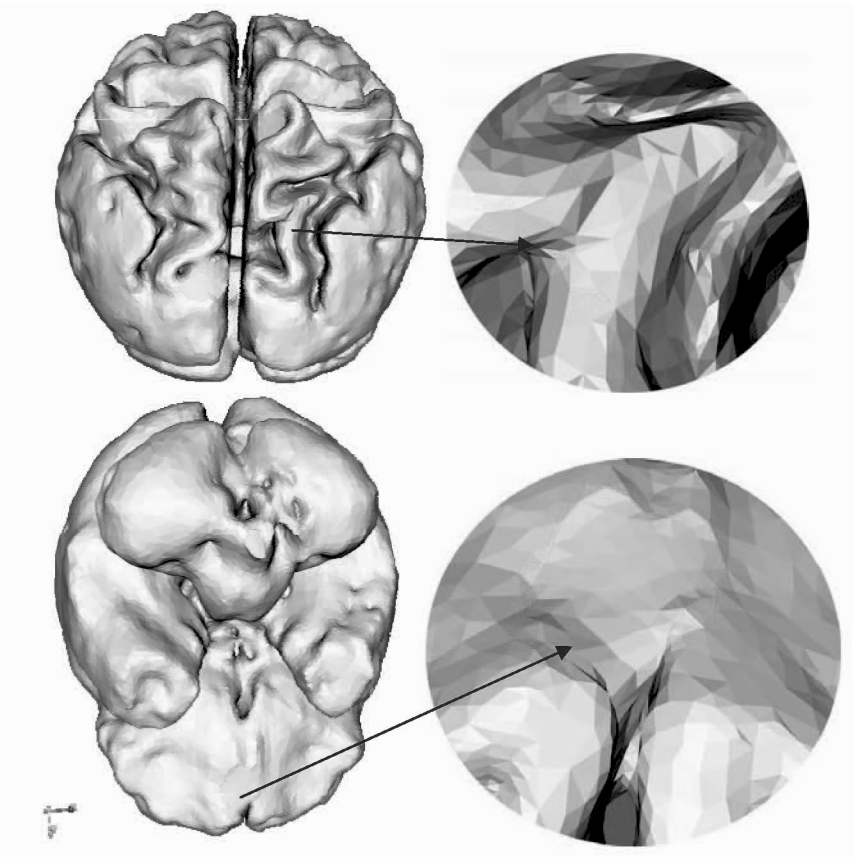


Figure 10. Patches for 13 holes on the white matter model

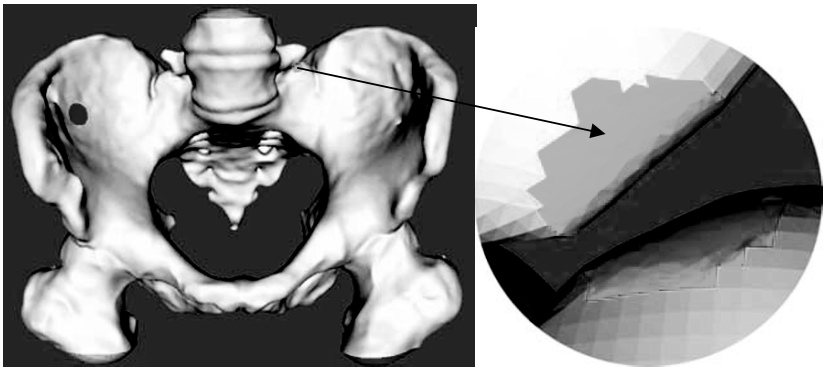


Figure 11. Human pelvis model with 10 holes patched on its surface

References

1. Barequet, G. and Sharir M., "Filling Gaps in the Boundary of a Polyhedron," *Computer Aided Geometric Design*, Vol. 12, 1995, pp. 207-229.
2. Barequet, G. and Kumar, S., "Repairing CAD Models," Proceedings of IEEE Visualization 1997, 1997, pp. 363-370.
3. Lee, Y. K., Lim, C. K., Ghazialam, H., Vardhan, H. and Eklund, E., "Surface Mesh Generation for Dirty Geometries by Shrink Wrapping using Cartesian Grid Approach," Proceedings of the 15th International Meshing Roundtable, Springer, 2006, pp. 393-410.
4. Jun, Y., "A Piecewise Hole Filling Algorithm in Reverse Engineering," *Computer-Aided Design*, Vol. 37, 2005, pp. 263-270.
5. 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, 2002, pp. 428-861.
6. Curless, B. and Levoy M., "A Volumetric Method for Building Complex Models from Range Images," *Computer Graphics*, Vol. 30, 1996, pp. 303-312.
7. Ju, T., "Robust Repair of Polygonal Models," Proceedings of ACM SIGGRAPH, 2004 ACM Transactions on Graphics, Vol. 23, pp. 888-895.
8. Liepa, P., "Filling Holes in Meshes," Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry processing, Eurographics Association, 2003, pp. 200-205.

9. Kobbelt, L. P., Vorsatz, J., Ulf, L. and Seidel, H.-P., "A Shrink Wrapping Approach to Remeshing Polygonal Surfaces," *Computer Graphics Forum (Eurographics '99)*, Vol. 18, 1999, pp. 119-130.
10. Bruno, L., "Dual Domain Extrapolation," *ACM Transactions on Graphics (SIGGRAPH)*, Vol. 22, 2003, pp. 364-369.
11. Branch, J., Prieto, F. and Boulanger, P., "A Hole-Filling Algorithm for Triangular Meshes using Local Radial Basis Function," Proceedings of the 15th International Meshing Roundtable, Springer, 2006, pp. 411-431.
12. Kumar, A., "Hole Patching in 3D Unstructured Surface Mesh," Masters Thesis, University of Alabama at Birmingham, Birmingham, AL, May 2007.
13. Piegel, L. A. and Tiller, W., *The NURBS Book*, 2nd ed., Springer, Berlin, 1996.
14. Lorenson, W. E. and Cline, H. E., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics*, Vol. 21, 1987, pp. 163-169.
15. Visualization ToolKit (VTK), http://www.vtk.org/.
16. Ito, Y., Shum, P. C., Shih, A. M., Soni, B. K. and Nakahashi, K., "Robust Generation of High-Quality Unstructured Meshes on Realistic Biomedical Geometry," *International Journal for Numerical Methods in Engineering*, Vol. 65, Issue 6, 2006, pp. 943-973.
17. Yoo, T. S., *Insight into Images. Principles and Practice for Segmentation, Registration, and Image Analysis*, A K Peters, Wellesley, MA, 2004.