Open Topology: A Toolkit for Brain Isosurface Correction

Sylvain Jaume[1], Patrice Rondao[2], and Benoît Macq[2]

[1] National Institute of Research in Computer Science and Control, INRIA, France, sylvain@mit.edu,
[2] Telecommunications and Remote Sensing Laboratory, University of Louvain, Belgium.

Abstract. Isosurface extraction from brain images often creates handles between brain folds that are anatomically separated. Although manual editing could optimally correct these anatomical errors, it is not realistic due the size of the 3D data and the convoluted geometry of the brain. We propose an algorithm to automatically repair the isosurface and we make our code available at http://www.OpenTopology.org.

Keywords: Isosurface, Marching Cubes, Topology Correction, Digital Topology, Handles, Brain Cortex.

1 Introduction

Anatomically the surface of the brain cortex is similar to a folded sheet. However its representation using an isosurface extracted from a typical brain image generally results in numerous handles. The correct visualization of the cortex for the patient and its flattening are therefore prevented.

Recently various works have studied the correction of the brain surface. We distinguish four major approaches: the image-based methods [1] [2], the graph-based methods [3] [4] [5], the surface-construction methods [6] [7], [8], and the surface-correction methods [9] [10] [11]. The image-based methods always provide a valid 3D object, and they offer a complete representation since an isosurface can be extracted from the corrected image. However some image methods using a region growing may reject large region when they hit a handle. The surface-based methods are generally faster, but occasional self-intersections are difficult to avoid during the correction. The graph-based methods and the level-set methods provide some interesting approaches, although often computationally intensive.

We propose an algorithm that combines the efficiency of the surface-based methods and the useful representation of the image-based methods. We have developed our code in C++ using the Visualization Toolkit [12] and have made the code [13] available to the Community. The paper is organized as follows. First in section 2 we define the theoretical background and present our method. Then in section 3 we visualize some results and assess the performance on ten brain datasets. Finally we conclude and give some perspectives in section 4.

2 Method

Since a 3D image is a discrete representation where the unit block is a voxel, we need to define the situation when two voxel are connected. If every voxel is a cube(1), we define one configuration, for two voxels to be connected, among the following three:

- two voxels share a face (6-connectivity),
- two voxels share a face or an edge (18-connectivity), or
- two voxels share a face, an edge or a corner (26-connectivity).

Once the connectivity of the foreground voxels is defined, the dual connectivity is assigned to the background voxels, or topological ambiguities may occur. We adopt the common 26-6 connectivity, i.e. foreground voxels are 26-connected while background voxels are 6-connected.

To extract a polygon surface from a 3D image of a brain, we use the Marching Cubes algorithm [14]. Then the number of handles of a polygon surface is derived from the Euler characteristic χ as formulated in Equation 1.

$$\chi = V - E + F = 2C - 2g \tag{1}$$

In this formula, V represents the number of vertices, E the number of edges, F the number of faces, C the number of connected components, and g the number of handles on the surface or holes in 3D. The number g, called the genus of the geometrical shape, is an invariant under the affine transforms.

Figure 1 shows the three main steps of our algorithm: the Marching Cubes, the handle detection, and the image correction. The Marching Cubes converts the image representation into a surface representation, required by our handle detection step. When the entire surface has been processed and every handle has been detected, the location of the handles is passed to the image correction step. Finally the image correction combines the input image and the location of the handles to generate the corrected image.

Figure 2 illustrates how a contour propagation can be used to detect handles. The detection of handles on a surface in 3D space is similar to the detection of the missing triangle in this example. The contour, highlighted in red in the left image, is initiated from the first triangle in the surface. Then it propagates from triangle to triangle as indicated in the middle image. Finally, when two parts of the contour have propagated around the missing triangle, they merge and therefore this event defines that a handle has been detected. The edge where the merging occurs is colored in yellow in the right image.

3 Results

Figure 3 indicates how a handle in a brain dataset is corrected. The red arrow in the left image points to an erroneous handle between the two hemispheres.

[1] In general, a voxel is a cube with a different edge length along every axis, but this has no impact for the Topology.

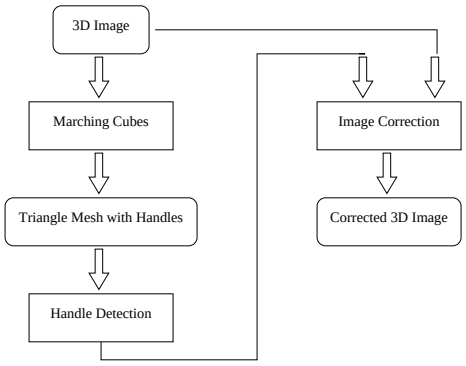

Fig. 1. The handle detection operates on the surface, but the correction is applied to the image. To obtain a surface, the user only needs to apply an additional Marching Cubes on the output image (not represented).

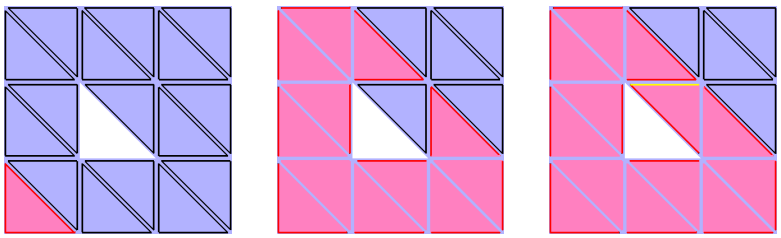

Fig. 2. To detect a handle (represented by the missing triangle), the red contour is propagated over the surface. As illustrated from the left image to right image, the contour propagates to neighbor triangles until two parts of the contour merge together. The edge in yellow in the right image highlights where the handle is detected.

To correct this handle, a small number of voxels are modified within the input image. The modified voxels are represented with red dots in the middle image. The isosurface extracted from the corrected image is rendered in the right image to demonstrate that the handle has been removed.

To test our algorithm, we corrected ten brain images with 256x256x124 voxels on a 1.6 GHz laptop with 512MB of RAM. The timing results are presented in Figure 4. Note that the dataset #9 is corrected relatively faster than the other datasets. The reason is that some border slices have been removed to isolate the brain in the input image. Across the datasets the corrected image was produced in less than two minutes.

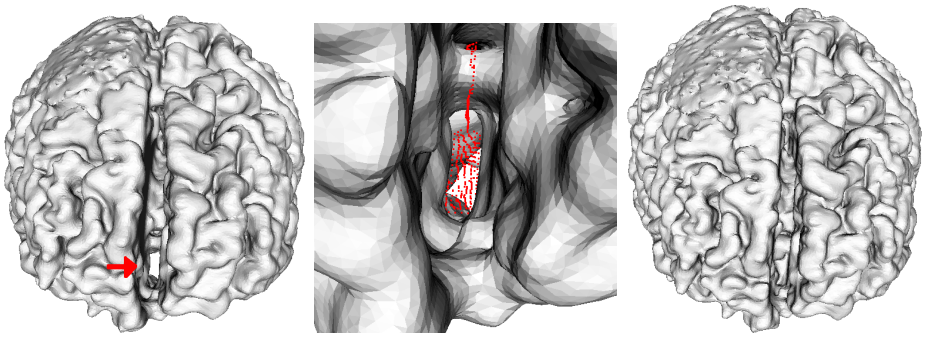

Fig. 3. The algorithm repairs the erroneous handle between the two brain hemispheres, shown by the red arrow. The red dots in the middle image represents the voxels that must be modified in the input image to remove the handle. The right image demonstrates that the handle has been effectively corrected.

Brain	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Genus	25	15	57	31	62	24	26	50	41	21
Marching	$17s$	$16s$	$17s$	$17s$	$16s$	$17s$	$17s$	$16s$	$13s$	$16s$
Detection	$29s$	$51s$	$47s$	$1m2s$	$36s$	$26s$	$1m15s$	$43s$	$20s$	$17s$
Correction	$9s$	$14s$	$23s$	$10s$	$18s$	$16s$	$7s$	$15s$	$3s$	$2s$
Total	$48s$	$1m21s$	$1m27s$	$1m29s$	$1m10s$	$59s$	$1m39s$	$1m14s$	$36s$	$35s$

Fig. 4. Our algorithm corrects every dataset in less than two minutes. The hole detection consumes most of the algorithm duration among the Marching Cubes, the hole detection and the handle correction. The first line indicates the genus, i.e. the number of handles, for every dataset.

4 Conclusion

We have presented a toolkit for the correction of brain isosurfaces. Our tests on 256x256x124 brain images have showed that our algorithm corrects a typical image in less than two minutes. Besides its efficiency, the algorithm generates both a corrected image and a corrected surface to visualize the brain of the patient. The Medical Community can download the source code described in this paper at http://www.OpenTopology.org Since topologically corrected surfaces can be compressed very efficiently and since various texture mapping methods require genus zero surfaces, we believe the proposed algorithm could find interesting applications in the Graphics Community.

References

1. Malandain, G., Bertrand, G., Ayache, N.: Topological segmentation of discrete surfaces. International Journal of Computer Vision **10** (1993) 183–197

2. Kriegeskorte, N., Goebel, R.: An efficient algorithm for topologically correct segmentation of the cortical sheet in anatomical mr volumes. NeuroImage **14** (2001) 329–46
3. Shattuck, D.W., Leahy, R.M.: Automated graph-based analysis and correction of cortical volume topology. IEEE Transactions on Medical Imaging **20** (2001) 1167–77
4. Han, X., Xu, C., Braga-Neto, U., Prince, J.: Topology correction in brain cortex segmentation using a multiscale graph-based algorithm. IEEE Transactions on Medical Imaging **21** (2002) 109–121
5. Ségonne, F., Fischl, B., Grimson, E.: Topology correction of subcortical structures. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). (2003)
6. Zeng, X., Staib, L.H., Schultz, R.T., Duncan, J.S.: Segmentation and measurement of the cortex from 3d mr images using coupled surfaces propagation. IEEE Transactions on Medical Imaging **18** (1999) 100–111
7. Bischoff, S., Kobbelt, L.: Isosurface reconstruction with topology control. In: Pacific Graphics Proceedings. (2002) 246–255
8. Bischoff, S., Kobbelt, L.: Topologically correct extraction of the cortical surface of a brain using level-set methods. Bildverarbeitung fr die Medizin (2004) 50–54
9. Fischl, B., Liu, A., Dale, A.M.: Automated manifold surgery: Constructing geometrically accurate and topologically correct models of the human cerebral cortex. IEEE Transactions on Medical Imaging **20** (2001) 70–80
10. Guskov, I., Wood, Z.: Topological noise removal. In: Graphics Interface. (2001) 19–26
11. Wood, Z.J., Hoppe, H., Desbrun, M., Schröder, P.: Removing excess topology from isosurfaces. ACM Transactions on Graphics **23** (2004)
12. Schroeder, W., Martin, K., Lorensen, B.: The Visualization Toolkit, An Object-Oriented Approach to 3D Graphics. 3rd edn. Kitware, Inc. (2002)
13. Jaume, S., Rondao, P., Macq, B.: Open source toolkit for brain isosurface correction. http://www.OpenTopology.org (2005)
14. Lorensen, W., Cline, H.: Marching cubes: A high resolution 3d surface construction algorithm. In: Computer Graphics. Volume 21 of 3. (1987) 163–169