Topologically Correct Extraction of the Cortical Surface of a Brain Using Level-Set Methods

Stephan Bischoff and Leif Kobbelt

Lehrstuhl für Informatik VIII, RWTH Aachen, 52062 Aachen
Email: {bischoff|kobbelt}@informatik.rwth-aachen.de

Abstract. In this paper we present a level-set framework for accurate and efficient extraction of the surface of a brain from MRI data. To prevent the so-called partial volume effect we use a topology preserving model that ensures the correct topology of the surface at all times during the reconstruction process. We also describe improvements that enhance its stability, accuracy and efficiency. The resulting reconstruction can then be used in downstream applications where we in particular focus on the problem of accurately measuring geodesic distances on the surface.

1 Introduction

In recent years, the problem of reconstructing the cortical surface of a brain from MRI data has received a good deal of attention [1], the main goals being:

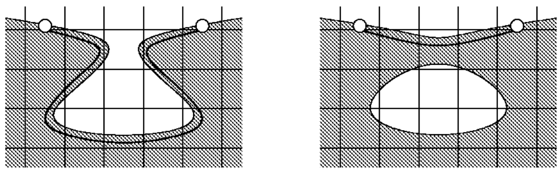
1. The reconstruction should closely fit to the measured data.
2. To prevent the partial volume effect, the topology of the reconstruction should match the topology of the brain itself.

In particular the second topic is of relevance if one wants to measure distances within the surface itself (geodesics). Functional regions for example are associated with the surface of the brain rather than with its interior. In this case topological errors in the reconstruction, although geometrically negligible, can lead to "short-cuts"from one part of the surface to another and will render the distance measurements useless, see Figure 1.

Deformable models have proven to be an effective tool to achieve the above goals. These models come in two flavors: Parametric models (often called snakes) represent the surface explicitly e.g. as a spline [2]. Level-set models (often called implicit models), on the other hand, represent the model as the isosurface of a scalar-valued function that is sampled on a Cartesian grid [3].

Because parametric models cannot change their topology, they have traditionally been favored over level-set models where the topology is difficult to control. Hence, once the surface is correctly initialized, a parametric model will keep its topology during the whole reconstruction process. On the downside, however, this comes at the expense of complex reparameterization strategies to avoid excessive internal stretch. Furthermore, parametric models require costly (self-)collision detection in each update step.

Fig. 1. Partial volume effect: Depending on the classification of the center voxel as interior or exterior, a small geometric inaccuracy can result in large differences in the distance of the two points.



In this work, we propose the use of level-set models for cortical surface reconstruction. These models provide the following advantages:

- They exhibit no parameterization artefacts and are always adequately sampled to the resolution of the underlying grid.
- The complex time- and space-continuous collision detection of parametric models is replaced with an efficient and robust discrete collision detection on a grid. In particular, a collision can only happen on grid edges and only when a grid point value changes its sign.

To prevent the partial volume effect, we make use of level-set models with built-in topology control [4]. However, we improve upon our previous work in several aspects that are described below.

2 Algorithm

2.1 Segmentation and surface extraction

For reconstructing the cortical surface we use a deformable model based on the topology preserving level set framework that was introduced in [4]. The basic idea is to represent the active contour as the zero level set of a scalar-valued function $f(x, y, z)$ that is sampled on the grid points $f_{ijk} = f(ih, jh, kh)$. The function f can be regarded as a signed distance to the contour, grid points ijk within the contour are called *conquered* and have negative f_{ijk}-values while outside grid points are positive. The algorithm proceeds by successively conquering grid points thereby expanding the contour. The order in which grid points are conquered is determined by *internal* as well as *external* forces that are derived from the intensity of the underlying MRI image. For each conquered grid point the algorithm checks whether the grid point is *complex*, i.e. whether it connects two previously unconnected components thereby creating a handle. If so, the topology change is resolved by assigning the grid point to one of the neighboring components and placing *cuts* on the edges to the other components, see Figure 2 for an illustration. When the contour has come to a halt, a variant of the Marching Cubes [5] algorithm is used to extract an explicit polygonal mesh representation of the contour. In the following sections we describe how to improve this basic algorithm with respect to stability, accuracy and efficiency.

Fig. 2. Cut edge grid: The conquered grid points $\circ, \diamond, \square, \triangle$ in configuration (a) locally make up three connected components that would incorrectly be connected by conquering the center grid point $\triangle$ in (b). To avoid this, the center point is assigned to one of the components and the edges to the other components are cut (c).

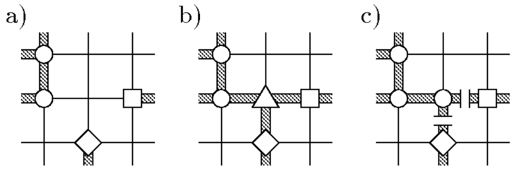


Fig. 3. Upwind scheme: To determine the cut edges we compute the upwind direction in each grid point (a) and then connect to the grid point with the most similar one (b). Compared to previous work [9] that shows strong bias towards the coordinate axes (c), the sub-voxel accuracy of the cut edge framework results in a better stability of the reconstruction (c).

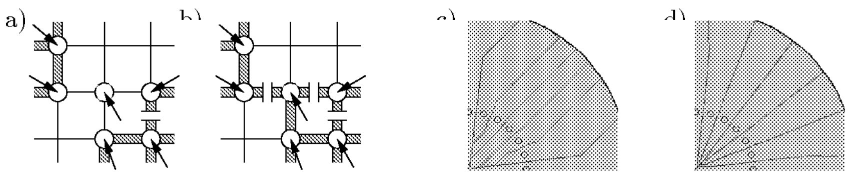


Fig. 4. Computing the cut location. In a post-processing step the two adjacent components P, Q of each cut edge $e = (p, q)$ are advanced separately across the cut interface (a, b and c). The location x of the cut is then computed from the intersection of the two height-profiles of the arrival times of P and Q resp. (d).

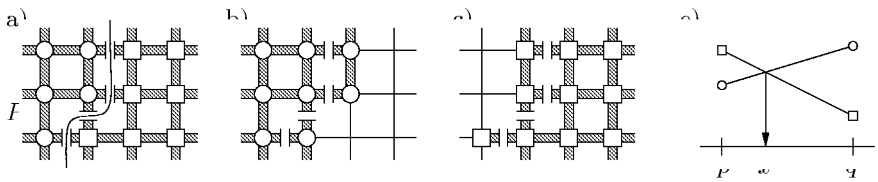
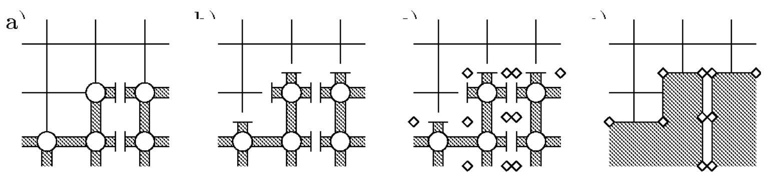


Fig. 5. Extracting the contour: Given a cut-edge grid (a) we first introduce virtual cuts between conquered and non-conquered voxels (b). In a second step we compute a sample point for each voxel that is adjacent to at least one cut (c). Finally we connect these sample points by quadrilaterals (d).



Initialization. We initialize the deformable model with the interface of the grey and white matter which is usually easy to segment in MRI data. To ensure the correct topology of this initial surface, we apply the algorithm of Kriegeskorte and Goebel [6].

Determining the cut edges. In the original formulation, a complex grid point is always connected to the nearest (in terms of arrival time of the contour) neighboring component. However, in a level-set framework, it is also possible to explicitly compute for each grid point the so-called *upwind direction*, i.e. the direction from where the contour arrives at the grid point. Comparing these upwind directions instead of the arrival times significantly improves the stability of the algorithm in the sense that it does not exhibit a directional bias towards the coordinate axes, see Figure 3 for an illustration.

Computing the cut location. In the original algorithm, the location of a cut on an edge $e = (p, q)$ is determined by extrapolating the arrival times at the grid points p, q adjacent to that edge. This leads to inaccuracies in particular if the direction of the edge is close to tangential to the contour. To improve upon this we proceed as follows (compare Figure 4): In a post-processing step, we locally advance each of the two connected components P, Q adjacent to the cut edge separately and then deduce the cut location from the corresponding arrival times.

Extracting a polygonal mesh. To extract a polygonal representation of the contour, the original algorithm uses a variant of the Marching Cubes algorithm that respects the cut edges and then locally applies a mesh decimation scheme. This algorithm is hard to implement efficiently and furthermore often produces unnecessarily many triangles. Hence we propose an extraction method that is similar to the dual contouring algorithm presented in [7]. Let us call the cube spanned by 8 grid points a *voxel*. First we introduce virtual cuts on edges connecting conquered and non-conquered grid points, i.e. edges that cross from the interior of the contour to the exterior. Then we collect all voxels that are adjacent to a cut-edge in a set V. For each voxel $v \in V$ we compute a sample point p_v as the average of the location of the cuts that are adjacent to v. Then we construct two opposing quadrilaterals for each cut-edge e by connecting the four sample points of the voxels adjacent to e (in the case of a virtual cut we only construct one quadrilateral), see Figure 5. If necessary, the resulting quadrangle mesh can then be triangulated and smoothed.

2.2 Geodesic measurement

Our system lets the user specify an arbitrary reference point on the reconstructed surface and then computes the geodesic distances from this point to all other points on the surface. This is done by an accurate level-set model which directly operates on triangulated manifolds [8].

Fig. 6. Geodesic distances. For visualization purposes the geodesic distances are color-coded as black and white stripes. Left: Reconstruction of a part of the cortical surface. Middle: Without topology control the sulci are not correctly reconstructed. Right: Using topology control the distance field correctly follows the sulci and gyri.

3 Results

We have found that our new update strategies for computing the collision points on the cut-edges increase the stability and accuracy of the algorithm significantly and compare favorably to the model proposed in [9], see Figure 3.

We have applied our algorithm to synthetic as well as to real MRI datasets. The running times for steps 1 and 2 of the algorithm are in the order of a few minutes for a typical $256 \times 256 \times 256$ dataset. The computation of the geodesic distances in step 3 is only a matter of seconds and allows for an efficient measuring and interactive exploration of the reconstructed surface. Figure 6 demonstrates these results.

References

1. Dale AM, Fischl B, Sereno MI: Cortical Surface-Based Analysis I: Segmentation and Surface Reconstruction. NeuroImage 9(2):179–94, 1999
2. Kass M, Witkin A, Terzopolous D: Snakes: Active Contour Models. International Journal of Computer Vision, 1:321–331, 1988
3. Osher S, Sethian JA: Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. J. Comput. Phys., 79(1):12–49, 1988
4. Bischoff S, Kobbelt L. Sub-Voxel Topology Control for Level-Set Surfaces. Computer Graphics Forum 22(3), 273–280, 2003
5. Lorensen WE, Cline HE. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. Computer Graphics, 21(4),163–169, 1987
6. Kriegeskorte N, Goebel R: An Efficient Algorithm for Topologically Correct Segmentation of the Cortical Sheet in Anatomical MR Volumes. NeuroImage 14, 329–346, 2001
7. Ju T, Lossaso F, Schaefer S, Warren J. Dual Contouring of Hermite Data. SIGGRAPH Conference Proceedings, 339–346, 2002
8. Novotni M, Klein R. Computing Geodesic Distances on Triangular Meshes. WSCG, 341–347, 2002
9. Han X, Xu C, Prince JL. A Topology Preserving Level Set Method for Geometric Deformable Models. IEEE TPAMI, 25(6):755–768, 2003