
Brain Mapping Using Topology Graphs Obtained by Surface Segmentation

Fabien Vivodtzev¹, Lars Linsen², Bernd Hamann², Kenneth I. Joy², and
Bruno A. Olshausen³

¹ Laboratoire GRAVIR (CNRS, INP Grenoble, INRIA, UJF).

`fabien.vivodtzev@imag.fr`

² Institute for Data Analysis and Visualization (IDAV), Department of Computer
Science, University of California, Davis. `{llinsen}@ucdavis.edu`,

`{hamann|joy}@cs.ucdavis.edu`

³ Center for Neuroscience, Department of Psychology, University of California,
Davis. `baolshausen@ucdavis.edu`

Summary. Brain mapping is a technique used to alleviate the tedious and time-consuming process of annotating brains by mapping existing annotations from brain atlases to individual brains. We introduce an automated surface-based brain mapping approach. After reconstructing a volume data set (trivariate scalar field) from raw imaging data, an isosurface is extracted approximating the brain cortex. The cortical surface can be segmented into gyral and sulcal regions by exploiting geometrical properties. Our surface segmentation is executed at a coarse level of resolution, such that discrete curvature estimates can be used to detect cortical regions. The topological information obtained from the surface segmentation is stored in a topology graph. A topology graph contains a high-level representation of the geometrical regions of a brain cortex. By deriving topology graphs for both atlas brain and individual brains, a graph node matching defines a mapping of brain cortex regions and their annotations.

1 Introduction

Annotating brains is a tedious and time-consuming process and can typically only be performed by an expert. A way to alleviate and accelerate the process is to take an already existing completely annotated brain and map its annotations onto other brains. The three-dimensional, completely annotated brain is called neuroanatomical brain atlas. An atlas represents a single brain or unified information collected from several “healthy” brains of one species. The digital versions of atlas brains are stored in databases [30]. Neuroscientists can benefit from this collected information by connecting to the database, accessing atlas brains, and mapping annotations onto their own data sets.

We propose an automated brain mapping approach that consists of several processing steps leading from three-dimensional imaging data to mapped cortical surfaces. Data sets are typically obtained in a raw format, which is the output of some imaging technique, such as functional magnetic resonance imaging (fMRI), given as a stack of aligned two-dimensional images. Isosurface extraction is used to obtain a surface representation of the brain cortex.

The shape of the brain cortex is complex, having many winding folds and creases, but its main characteristic can be described by alternating convex and concave regions called *gyri* and *sulci*. We use a multiresolution surface representation, since fine details are not present at a coarse level of resolution, while more global gyral and sulcal brain structures still are. We detect gyri and sulci by exploiting discrete curvature estimates, and segment the cortical surface into distinct regions based on curvature. The curvature-based segmentation is independent of the size of the segments. Thus, it is capable of extracting cortical regions of varying size and of detecting corresponding cortical regions in atlas and user brains, even if a region's size varies substantially for two brains being compared.

Curvature-based segmentation leads to a topological characterization of the surface. A topology graph is used to store the topological surface information at a high level. The brain mapping is executed by generating topology graphs for atlas brain and user brains and finding node correspondences for the graphs, where each node represents a cortical region.

2 Related Work

First approaches in brain mapping used rigid models and spatial distributions. In [26], a stereotactic atlas is expressed in an orthogonal grid system, which is rescaled to a patient brain, assuming one-to-one correspondences of specific landmarks. Similar approaches are discussed in [2, 5, 11] using elastic transformations. The variation in brain shape and geometry is of significant extent between different individuals of one species. Static rigid models are not sufficient to describe appropriately such inter-subject variabilities.

Deformable models were introduced as a means to deal with the high complexity of brain surfaces by providing atlases that can be elastically deformed to match a patient brain. Deformable models use snakes [20], B-spline surfaces [24], or other surface-based deformation algorithms [8, 9]. Feature matching is performed by minimizing a cost function, which is based on an error measure defined by a sum measuring deformation and similarity. The definition of the cost function is crucial. Some approaches rely on segmentation of the main sulci guided by a user [4, 27, 29], while others automatically generate a structural description of the surface.

Level set methods, as described in [21], are widely used for convex shapes. These methods, based on local energy minimization, achieve shape recognition requiring little known information about the surface. Initialization must be

done close to surface boundaries, and interactive seed placement is required. Several approaches have been proposed to perform automatically the seeding process and adapt the external propagation force [1], but small features can still be missed. Using a multiresolution representation of the cortical models, patient and atlas meshes are matched progressively by the method described in [16]. Folds are annotated according to size at a given resolution. The choice of the resolution is crucial. It is not guaranteed that same features are present at the same resolution for different brains.

Many other automatic approaches exist, including techniques using active ribbons [13, 10], graph representations [3, 22], and region growing [18]. A survey is provided in [28]. Even though some of the approaches provide good results, the highly non-convex shape of the cortical surface, in combination with inter-subject variability and feature-size variability, leads to problems and may prevent a correct feature recognition/segmentation and mapping without user intervention.

Our approach is an automated approach that can deal with highly non-convex shapes, since we segment the brain into cortical regions, and with feature-size as well as inter-subject variability, since it is based on discrete curvature behavior. Moreover, isosurface extraction, surface segmentation, and topology graphs are embedded in a graphical system supporting visual understanding.

3 Brain Mapping

Our brain mapping approach is based on a pipeline of automated steps. Figure 1 illustrates the sequence of individual processing steps.

The input for our processing pipeline is discrete imaging data in some raw format. Typically, imaging techniques produce stacks of aligned images. If the images are not aligned, appropriate alignment tools must be applied [25]. Volumetric reconstruction results in a volume data set, a trivariate scalar field.

Depending on the used imaging technique, a scanned data set may contain more or less noise. We mainly operate on fMRI data sets, thus having to deal with significant noise levels. We use a three-dimensional discrete Gaussian smoothing filter, which eliminates high-frequency noise without affecting visibly the characteristics of the three-dimensional scalar field. The size of the Gaussian filter must be small. We use a $3 \times 3 \times 3$ mask locally to smooth every value of a rectilinear, regular hexahedral mesh. Figure 2 shows the effect of the smoothing filter applied to a three-dimensional scalar field by extracting isosurfaces from the original and filtered data set.

After this preprocessing step, we extract the geometry of the brain cortex from the volume data. The boundary of the brain cortex is obtained via an isosurface extraction step, as described in Section 4. If desired, isosurface extraction can be controlled and supervised in a fashion intuitive to neuroscientists.

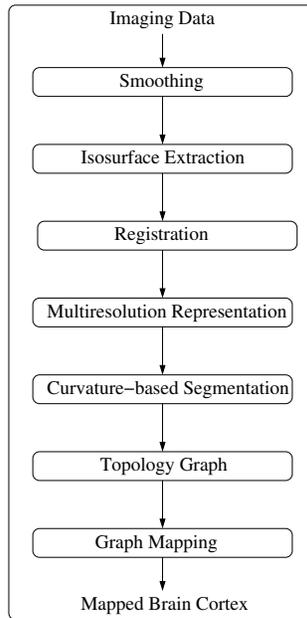
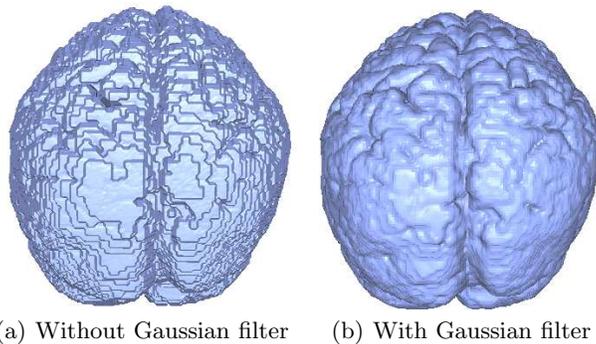


Fig. 1. Processing pipeline: from imaging data in raw format to mapped brain cortex surface.



(a) Without Gaussian filter (b) With Gaussian filter

Fig. 2. Smoothing of volumetric scalar field visualized by extracting isosurfaces from original and filtered data.

Once the geometry of the brain cortices is available for both atlas brain and a user brain, the two surfaces can be registered. Since our brain mapping approach is feature-based, we perform the registration step by a simple and fast rigid body transformation. For an overview and a comparison of rigid body transformation methods, we refer to [6].

User-guided surface segmentation of the brain cortices is based on curvature estimates. Since curvature estimates are sensitive to high-frequency de-

tail, a multiresolution approach is used, as described in Section 5. On a coarse level of resolution, only the main (low-frequency) features of the brain cortices are represented while the small (high-frequency) details are not present.

Curvature estimates on surfaces are used to distinguish between regions of different behavior [12]. We use Gaussian and mean curvatures to distinguish between elliptic and hyperbolic regions and between convex and concave regions, respectively. The shape of a brain cortex is mainly defined by gyral and sulcal regions. We segment the surface based on these curvature characteristics, as described in Section 6.

Curvature-based surface segmentation describes the topological behavior of the surface, which we store in a topology graph, as described in Section 7. Nodes of the topology graph represent regions of the cortical surface. Neighborhood information of such regions is represented by edges in the graph.

The final brain mapping is performed on the high-level and abstract representation of a topology graph, as described in Section 8. We construct a topology graph for the atlas brain and a user brain and determine matching node correspondences.

4 Isosurface Extraction

Extracting the geometry of a brain cortex from a discrete trivariate scalar field can be done by standard isosurfacing techniques. We decided to use a marching cubes-like approach [19]. For the quality of brain mapping it is crucial to choose an “appropriate isovalue,” such that the extracted isosurface follows closely the geometrical shape of the brain cortex.

To validate the proper choice of an isosurface, we designed a tool that allows a user to supervise the isosurfacing procedure. Traditionally, neuroscientists segment data slice-by-slice in a two-dimensional setup. Thus, the supervision tool should allow them to inspect the original two-dimensional slices and an extracted segment boundary for that particular slice simultaneously.

Figure 3 shows an example of supervised isosurface extraction. The upper row shows isosurfaces extracted for various isovalues. The two rows below show two original two-dimensional slices with overlaid cross sections (red contour) of the extracted isosurface. The left column shows the location of the slice with respect to the isosurface. In this particular example, the chosen isovalue is a good one, since the red contours follow closely the gyri and sulci of the brain cortex.

Due to remaining noise in the data set, the isosurface extraction step produces one large main component and many small isolated components. The main component represents the brain cortex, while the small components should be removed. We use a surface-growing algorithm that generates a watertight triangular mesh in a half-edge data structure representing the largest component. The small components are removed.

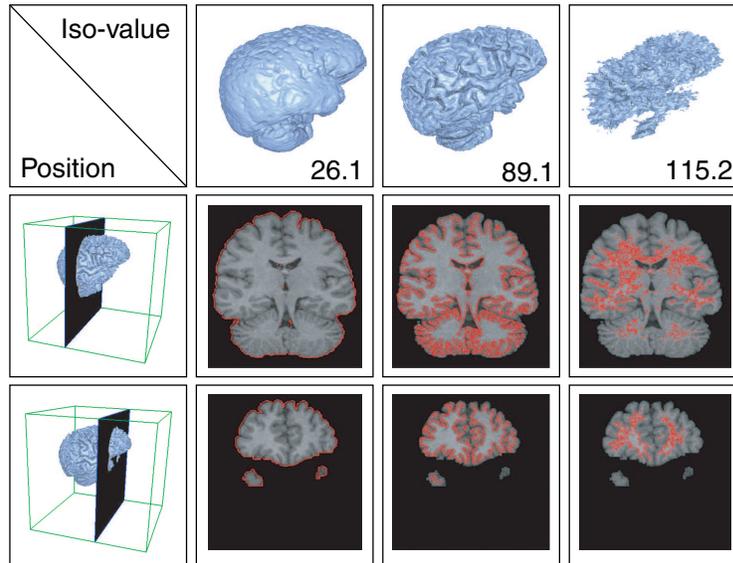


Fig. 3. Supervised isosurface extraction: overlaying original two-dimensional slices with cross sections (red contour) of extracted isosurface.

5 Multiresolution Surface Representation

To obtain a multiresolution surface representation of a brain cortex, we start with the triangular isosurface mesh. To simplify the high-resolution triangular mesh we use a simplification algorithm based on progressive meshes [14]. We iteratively apply edge-collapse operations. Although collapsing an edge is a simple operation, it can modify topology and geometry. To ensure consistency of our mesh, we use consistency checks as described in [15], based on topological analysis in the neighborhood affected by a collapse operation.

For each edge of the mesh, an error corresponding to the cost of its collapse is computed and stored. According to this value an ordered heap of edges is created. During mesh simplification, the method identifies the top edge, checks for consistency, and, if possible, collapses it. This process is highly dependent on the error metric used to decide which edge to collapse next. Many metrics have been proposed for edge collapse algorithms over the past decade [7, 14, 17, 23]. Most of these metrics attempt to preserve “sharp” edges and details. Our objective, instead, is to remove detail even in regions of high curvature. Thus, our error metric is only based on edge length, and our main goal is to create a near-uniform distribution of vertices on the surface. After a valid collapse, the affected neighborhood is updated accordingly.

Topology (i. e., adjacency information of triangles) and geometry (i. e., positional information of the resulting “collapse” vertex) are modified by an edge collapse. An edge collapses to its midpoint. (We decided not to optimize

the position to keep computation costs low.) Using midpoints also reduces the risk of self-intersections.

Figure 4 shows the result of simplifying a triangular mesh. The main gyral and sulcal features of the cortical surface are well preserved at the coarse level of resolution.

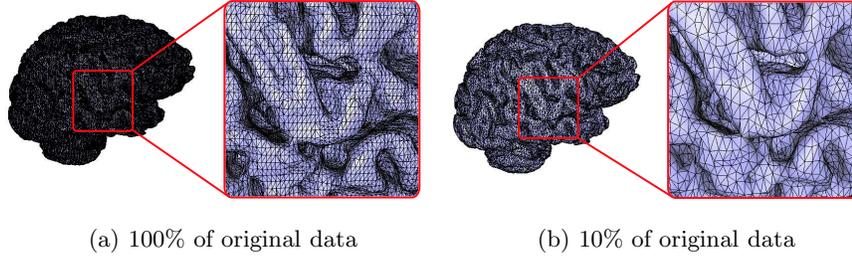


Fig. 4. Multiresolution surface representation.

6 Surface Segmentation

6.1 Curvature-based Surface Characteristics

A surface can be divided into regions of elliptic and hyperbolic behavior. The regions of elliptic behavior can further be classified into convex and concave regions. When considering the a brain cortex, the gyri contain convex elliptic regions and the sulci contain concave elliptic regions. The blending areas between gyri and sulci are hyperbolic regions. This observation led to our decision to use curvature-based surface characteristics for user-guided surface segmentation. Discrete curvature estimates and their use for curvature-based surface segmentation were introduced in [31].

We use mean curvature estimates to distinguish between convex and concave regions. A discrete version of the mean curvature operator at a vertex \mathbf{x}_i of a triangular mesh can be defined by the length of a vector operator $\mathbf{K}(\mathbf{x}_i)$. For characterizing surface behavior with respect to mean curvature, we only need to use the direction of $\mathbf{K}(\mathbf{x}_i)$. Thus, we use a simplified operator $\mathbf{K}_{dir}(\mathbf{x}_i)$. The vector $\mathbf{K}_{dir}(\mathbf{x}_i)$ associated with a vertex \mathbf{x}_i is computed as a weighted sum of difference vectors emanating from \mathbf{x}_i and ending at the vertices being edge-connected with \mathbf{x}_i . The weight of the vector associated with edge e_{ij} between \mathbf{x}_i and its neighbor \mathbf{x}_j depends on the cotangents taken from the opposite angles of its adjacent faces. This operator is defined as

$$\mathbf{K}_{dir}(\mathbf{x}_i) = \sum_{j=1}^{N_i} (\cot \alpha_j + \cot \beta_j) (\mathbf{x}_j - \mathbf{x}_i) ,$$

where N_i is the number of neighbors constituting the set of edge-connected neighbor vertices of \mathbf{x}_i , and α_j, β_j are the opposite angles of e_{ij} with respect to its adjacent faces, see Figure 5.

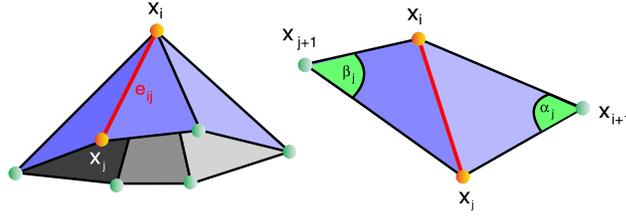


Fig. 5. Parameters used by mean curvature operator.

We use the operator $\mathbf{K}_{dir}(\mathbf{x}_i)$ to define the Boolean operator $mean(\mathbf{x}_i)$, which allows us to distinguish between convex and concave regions. It is defined as

$$mean(\mathbf{x}_i) = \begin{cases} \text{convex} & \text{if } \mathbf{K}_{dir}(\mathbf{x}_i) \cdot \mathbf{n}_i \leq 0 \\ \text{concave} & \text{if } \mathbf{K}_{dir}(\mathbf{x}_i) \cdot \mathbf{n}_i > 0 \end{cases}$$

where \mathbf{n}_i is a discrete approximation of the normal vector at \mathbf{x}_i . In concave areas, the operator $\mathbf{K}_{dir}(\mathbf{x}_i)$ and the normal vector \mathbf{n}_i are directed in roughly opposite directions, whereas in convex areas they are directed in roughly the same direction, see Figure 6.

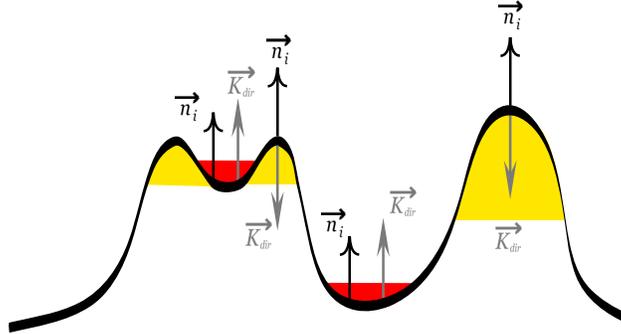


Fig. 6. Using mean curvature to distinguish between convex and concave regions.

To further distinguish between elliptic and hyperbolic regions, i. e., to separate local extrema from blending regions, we consider Gaussian curvature. A discrete version of the Gaussian curvature at a vertex \mathbf{x}_i of a triangular mesh can be defined by the length of an operator $\kappa_G(\mathbf{x}_i)$. This operator compares 2π with the sum of inner angles θ_j of all the adjacent faces of a vertex \mathbf{x}_i ,

see Figure 7. In the planar case, the sum of the angles is 2π . When \mathbf{x}_i is an extremum, a plane through \mathbf{x}_i exists, such that all neighbor vertices of \mathbf{x}_i lie on one side of that plane, see Figure 7. Thus, the angles sum to a value smaller than 2π . When \mathbf{x}_i is not an extremum and we compute the best fitting plane in the least-squares sense through \mathbf{x}_i , the neighbor vertices lie above and below that plane. In this situation, the angles sum up to a value larger than 2π . Hence, we consider only the sign of the operator $\kappa_G(\mathbf{x}_i)$, defined as

$$\kappa_G(\mathbf{x}_i) = 2\pi - \sum_{j=1}^{N_i} \theta_j ,$$

where θ_j is the angle between the difference vectors $\mathbf{x}_j - \mathbf{x}_i$ and $\mathbf{x}_{j+1} - \mathbf{x}_i$, emanating from vertex \mathbf{x}_i and ending at neighbors \mathbf{x}_j and \mathbf{x}_{j+1} , respectively, see Figure 7.

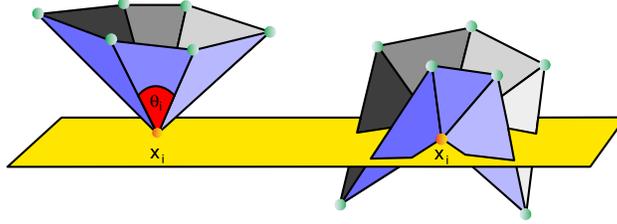


Fig. 7. Using Gaussian curvature to distinguish between elliptic and hyperbolic regions.

We use $\kappa_G(\mathbf{x}_i)$ to define another Boolean operator $Gauss(\mathbf{x}_i)$, which is true if the vertex \mathbf{x}_i is a local extremum:

$$Gauss(\mathbf{x}_i) = \begin{cases} \text{elliptic} & \text{if } \kappa_G(\mathbf{x}_i) > 0 \\ \text{hyperbolic} & \text{if } \kappa_G(\mathbf{x}_i) \leq 0 \end{cases} .$$

6.2 Curvature-based Segmentation

By combining the operators *mean* and *Gauss*, we can generate an initial surface segmentation consisting of regions of the same type of curvature. Figure 8 shows how a cortical surface of a human brain is partitioned into elliptical convex regions (yellow), elliptical concave regions (red), hyperbolic convex regions (green), and hyperbolic concave regions (blue).

Figure 9 shows the effect of the multiresolution surface representation by segmenting the high- and low-resolution surfaces from Figure 4 with respect to mean curvature. Only for the low-resolution surface, the cortical regions are detected as desired.

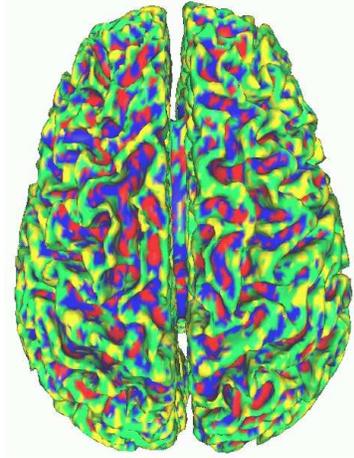


Fig. 8. Curvature-based segmentation of cortical surface.

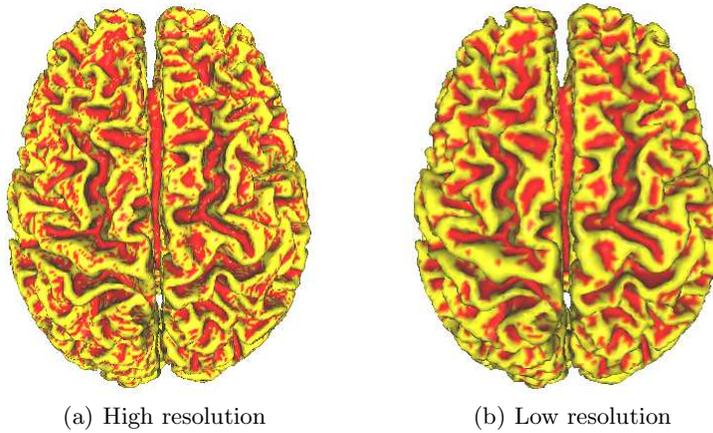


Fig. 9. Surface segmentation at different levels of resolution.

7 Topology Graph

Curvature-based surface segmentation implies a topology for the surface. We construct a graph that stores the topology information. The nodes in the graph represent regions of a certain curvature type, and the edges in the graph represent neighborhood information of the surface regions.

For cortical surfaces, gyral regions cover larger parts of the brain. Their segmentation into smaller functional regions cannot be done automatically, since it is not based on geometrical properties. Sulcal regions instead remain local. Thus, we decided to use sulci only for the construction of topology graphs.

Each node in the topology graph represents one sulcus. The node representing a certain sulcus is generated by collapsing all vertices of the triangulated surface that are characterized by the surface segmentation procedure as belonging to that sulcus. The position of the node is determined by averaging the positions of the collapsed vertices.

To determine neighborhood information for sulci on the cortical surface, we use a contour-growing algorithm. Starting from a polygonal contour that describes the boundary of a sulcus on the triangulated surface, we grow the contour iteratively by one triangle in all directions, i. e., after one iteration step, the new contour encloses all vertices of the old contour plus all its neighbors. If the contour of a sulcus, when growing, intersects another sulcus, then these sulci are considered neighbors, and the nodes representing these sulci in the topology graph are connected by an edge. The number of iteration steps depends on the resolution of the triangulated surface.

Figure 10 shows the generation of a topology graph for a cortical surface of a human brain. Figure 10(a) shows the segmented surface, where the detected sulci are rendered using random colors. Figure 10(b) shows the topology graph generated from the segmented surface, where nodes are shown in red and edges in blue. We applied surface segmentation at a resolution of 50%, and used four iteration steps for generating the edges in the graph.

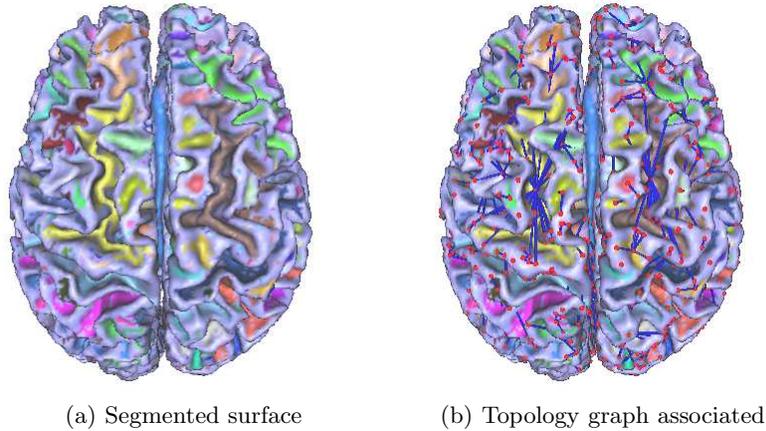


Fig. 10. Topology graph from surface segmentation.

8 Graph Mapping

We prepare the brain mapping step by generating a topology graph representation for both atlas brain and a user brain. The mapping is performed

by matching graph nodes. In addition to node position and edge connectivity information, the graphs also store, for each node, the size of the associated sulcus. Since edges in the triangular mesh have nearly the same length, the size of a sulcus can be estimated well by the number of vertices of the triangular mesh that are classified to belong to the sulcus.

For each node \mathbf{n}_u of the topology graph representing a user brain, we identify a node \mathbf{n}_a in the topology graph representing the atlas brain that provides a best match in terms of location and size. To find a best match for \mathbf{n}_u , we search for a node \mathbf{n}_a , representing the sulcus, whose size is closest to the size of the sulcus represented by \mathbf{n}_u . The search is restricted by limiting the Euclidean distance from \mathbf{n}_u and the topological distance in the graph to not being beyond a certain threshold.

Figure 11 shows the result of a graph-based brain mapping. Figure 11(a) shows the atlas brain, and Figure 11(b) shows the user brain. Colors of the sulci indicate which sulci of the atlas brain are associated with which sulci of the user brain. Regions consisting of less than a certain number of vertices are not considered as being useful and are not mapped (indicated by red in Figure 11(b)).

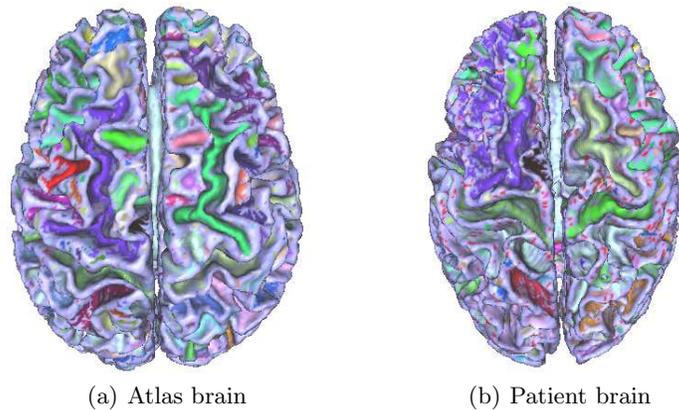


Fig. 11. Brain mapping based on topology graphs.

9 Conclusions and Future Work

We have presented an automated approach for brain mapping to map annotations of the cortical surface from a brain atlas to individual brains. After reconstructing trivariate scalar fields from raw imaging data, isosurfaces are extracted approximating brain cortices. A cortical surface is segmented into

gyral and sulcal regions by exploiting geometrical properties. Our surface segmentation step is performed at a coarse level of resolution, such that discrete curvature estimates can be used to detect cortical regions. The topological information obtained from the surface segmentation step is stored in a topology graph. A topology graph contains a high-level representation of the geometrically distinct regions of a brain cortex. By deriving topology graphs for both atlas brain and user brain, a high-quality brain mapping is obtained by mapping graph nodes.

We plan to extend the node matching process in a way that further exploits region neighborhood information. Moreover, we would like to develop a more sophisticated registration method, which we should lead to further improvement of node matching results.

Acknowledgments

This work was supported by the National Science Foundation under contracts ACI 9624034 (CAREER Award) and ACI 0222909, through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, and through the National Partnership for Advanced Computational Infrastructure (NPACI); the National Institute of Mental Health and the National Science Foundation under contract NIMH 2 P20 MH60975-06A2; and the Lawrence Livermore National Laboratory under ASCI ASAP Level-2 Memorandum Agreement B347878 and under Memorandum Agreement B503159. We thank the members of the Visualization and Graphics Research Group at the Center for Image Processing and Integrated Computing (CIPIC) at the University of California, Davis.

References

1. C. Baillard and C. Barillot. Robust 3d segmentation of anatomical structures with level sets. In G.-P. Bonneau, S. Hahmann, and Charles D. Hansen, editors, *Proceedings of Medical Image Computing and Computer-Assisted Intervention, MICCAI'00, LNCS 1935*, pages 236–245, 2000.
2. F.L. Bookstein. Thin-plate splines and the atlas problem for biomedical images. In A. Colchester and D. Hawkes, editors, *12th Internat. Conf. Information Processing in Medical Imaging, vol. 511 of Lecture Notes in Computer Science*, pages 326–342, 1991.
3. P. Cachier, J.F. Mangin, X. Pennec, D. Rivière, D. Papadopoulos-Orfanos, J. Regis, and N. Ayache. Multisubject non-rigid registration of brain MRI using intensity and geometric features. In W.J. Niessen and M.A. Viergever, editors, *4th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI'01), vol. 2208 of Lecture Notes in Computer Science*, pages 734–742, 2001.

4. D.L. Collins, G. Le Goualher, and A.C. Evans. Non-linear cerebral registration with sulcal constraints. In *First International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), LNCS 1496*, pages 974–984, 1998.
5. D.L. Collins, T.M. Peters, and A.C. Evans. An automated 3d nonlinear image deformation procedure for determination of gross morphometric variability in human brain. In *Proc. Conf. Visualization in Biomedical Computing, SPIE 2359*, pages 180–190, 1994.
6. D.W. Eggert, A. Lorusso, and R.B. Fisher. Estimating 3-d rigid body transformation: a comparison of four major algorithms. *Machine Vision and Applications*, 9:272–290, 1997.
7. M. Garland and P.S. Heckbert. Surface simplification using quadric error metrics. *Computer Graphics, 31st Annual Conference Series*, pages 209–216, 1997.
8. R.D. Rabbitt G.E. Christensen and M.I. Miller. 3-d brain mapping using a deformable neuroanatomy. *Physics in Medicine and Biology*, 39:609–618, 1994.
9. J.C. Gee, M. Reivich, and R. Bajcsy. Elastically deforming 3-d atlas to match anatomical brain images. *Journal of Computer Assisted Tomography*, 17:225–236, 1993.
10. G. Le Goualher, E. Procyk, L. Collins, R. Venegopal, C. Barillot, and A. Evans. Automated extraction and variability analysis of sulcal neuroanatomy. *IEEE Transactions on Medical Imaging, TMI*, 18(3):206–217, 1999.
11. T. Greitz, C. Bohm, S. Holte, and L. Eriksson. A computerized brain atlas: Construction, anatomical content, and some applications. *Journal of Computer Assisted Tomography*, 15:26–38, 1991.
12. B. Hamann. Curvature approximation for triangulated surfaces. In G. Farin, H. Hagen, and H. Noltemeier, editors, *Geometric Modelling, Computing Suppl. 8*, pages 139–153. Springer-Verlag, 1993.
13. P. Hellier and C. Barillot. Coupling dense and landmark-based approaches for non-rigid registration. *IEEE Transactions on Medical Imaging*, 22:974–984, 2003.
14. H. Hoppe. Progressive meshes. In *Proceedings of SIGGRAPH 1996*, pages 99–108. ACM Press, 1996.
15. H. Hoppe, T.D. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of SIGGRAPH 1993*, pages 19–26. ACM Press, 1993.
16. S. Jaume, B. Macq, and S.K. Warfield. Labeling the brain surface using a deformable multiresolution mesh. In *Proceedings of Medical Image Computing and Computer-Assisted Intervention, MICCAI 2002*, pages 451–458, 2002.
17. P. Lindstrom and G. Turk. Fast and efficient polygonal simplification. In *Proceedings of IEEE Conference on Visualization 1998*, pages 279–286. IEEE Computer Society Press, 1998.
18. G. Lohmann and D.Y. von Cramon. Automatic labeling of the human cortical surface using sulcal basins. *Medical Image Analysis*, 4(3):179–188, 2000.
19. William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH 1987*, pages 163–169. ACM Press, 1987.
20. A. Witkin M. Kass and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

21. R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on PAMI*, 17(2):158–175, 1995.
22. D. Rivière, J.F. Mangin, D. Papadopoulos, J.M. Martinez, V. Frouin, and J. Regis. Automatic recognition of cortical sulci using a congregation of neural networks. In *Third International Conference on Medical Robotics, MICCAI'00, Imaging and Computer Assisted Surgery*, pages 40–49, 2000.
23. R. Ronfard and J. Rossignac. Full-range approximation of triangulated polyhedra. *Computer Graphics Forum, Proceedings of Eurographics 1996*, 15(3), 1996.
24. S. Sandor and R. Leahy. Surface-based labeling of cortical anatomy using a deformable atlas. *IEEE Transaction on Medical Imaging*, 16(1):41–54, 1997.
25. D. Shulga and J. Meyer. Aligning large-scale medical and biological data sets: Exploring a monkey brain. In *Visualization, Imaging and Image Processing (VIIP 2001)*, pages 434–439. The International Association of Science and Technology for Development (IASTED), 2001.
26. J. Talairach and P. Tournoux. *Co-Planar Stereotaxic Atlas of the Human Brain, 3-Dimensional Proportional System: An Approach to Cerebral Imaging*. Thieme Medical Publisher, Inc., Georg Thieme Verlag, 1988.
27. P. Thompson and A.W. Toga. Detection, visualization and animation of abnormal anatomic structure with a deformable probabilistic brain atlas based on random vector field transformation. *Medical Image Analysis*, 1(2):271–294, 1996.
28. P.M. Thompson, R.P. Woods, M.S. Mega, and A.W. Toga. Mathematical/computational challenges in creating deformable and probabilistic atlases of the human brain. *Human Brain Mapping*, 9:81–92, 2000.
29. M. Vailland and C. Davatzikos. Hierarchical matching of cortical features for deformable brain image registration. In *Proceeding of IPMI'99, LNCS 1613, Springer-Verlag, Berlin, Germany*, pages 182–195, 1999.
30. D.C. Van Essen, J. Harwell, D. Hanlon, and J.P.M. Dickson. Surface-based atlases and a database of cortical structure and function. In S.H. Koslow and S. Subramaniam, editors, *Databasing the Brain: From Data to Knowledge (Neuroinformatics)*, John Wiley & Sons, 2003.
31. Fabien Vivodtzev, Lars Linsen, Georges-Pierre Bonneau, Bernd Hamann, Kenneth I. Joy, and Bruno A. Olshausen. Hierarchical isosurface segmentation based on discrete curvature. In G.-P. Bonneau, S. Hahmann, and Charles D. Hansen, editors, *Proceedings of VisSym '03, Eurographics-IEEE TVCG Symposium on Visualization*, 2003.

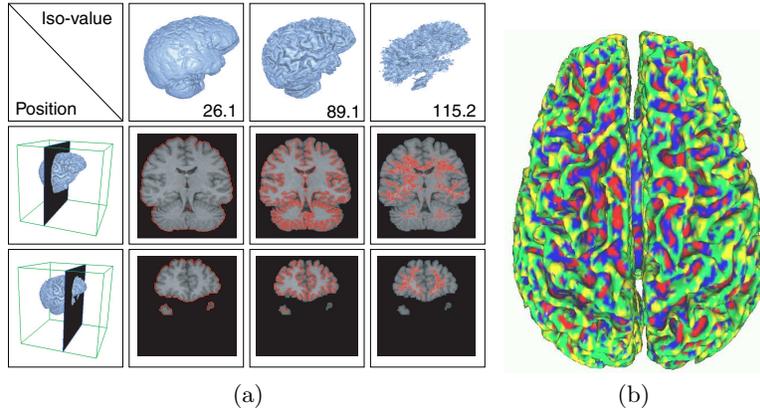


Fig. 12. (a) Supervised isosurface extraction: overlaying original two-dimensional slices with cross sections (red contour) of extracted isosurface. (b) Curvature-based segmentation of cortical surface.

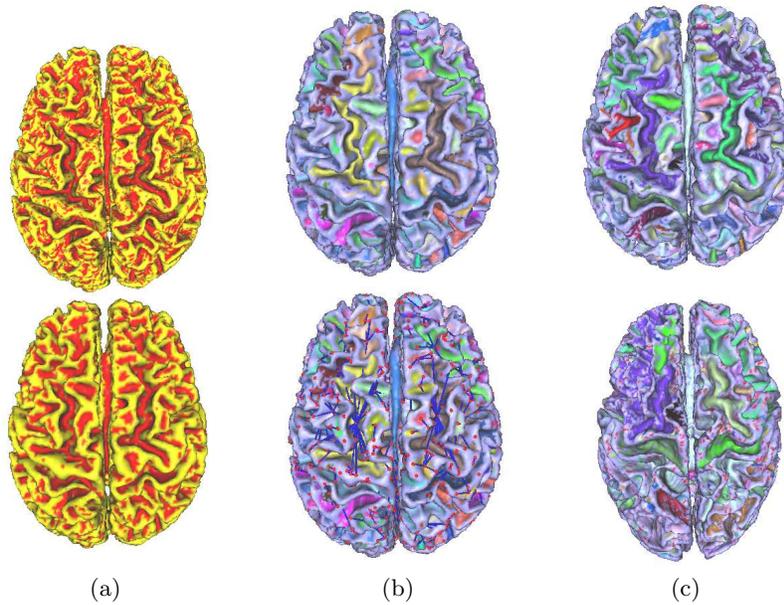


Fig. 13. (a) Surface segmentation at high resolution (*top*) and low resolution (*bottom*). (b) Segmented surface (*top*) and topology graph associated (*bottom*). (c) Brain mapping based on topology graphs in between an atlas brain (*top*) and a patient brain (*bottom*).