

Joint Sulcal Detection on Cortical Surfaces with Graphical Models and Boosted Priors

Yonggang Shi, Zhuowen Tu, Allan L. Reiss, Rebecca A. Dutton, Agatha D. Lee,
Albert M. Galaburda, Ivo Dinov, Paul M. Thompson, Arthur W. Toga*

Abstract—In this paper, we propose an automated approach for the *joint* detection of major sulci on cortical surfaces. By representing sulci as nodes in a graphical model, we incorporate Markovian relations between sulci and formulate their detection as a *maximum a posteriori* (MAP) estimation problem over the joint space of major sulci. To make the inference tractable, a sample space with a finite number of candidate curves is automatically generated at each node based on the Hamilton-Jacobi skeleton of sulcal regions. Using the AdaBoost algorithm, we learn both individual and pairwise shape priors of sulcal curves from training data, which are then used to define potential functions in the graphical model based on the connection between AdaBoost and logistic regression. Finally belief propagation is used to perform the MAP inference and select the joint detection results from the sample spaces of candidate curves. In our experiments, we quantitatively validate our algorithm with manually traced curves and demonstrate the automatically detected curves can capture the main body of sulci very accurately. A comparison with independently detected results is also conducted to illustrate the advantage of the joint detection approach.

Index Terms—Cortex, major sulci, shape prior, graphical model, AdaBoost, boosted prior

I. INTRODUCTION

One of the most intriguing and difficult problems in brain imaging is identifying and registering the convolution patterns of the cortex. It is generally agreed that a set of major sulci are relatively stable [1] and they have been used as landmark curves for registration and locating structural and functional areas on cortices [2], [3]. On the other hand, the automated detection of these sulci is still a challenging problem due to the complexity and variability of the convolution patterns and the different forms these sulci may have in the folding patterns. Thus, manual annotation remains the gold standard in brain mapping practice. In this paper, we propose a novel approach

for the *joint* detection of major sulci via the solution of an inference problem on graphical models [4], which we construct with boosting techniques [5] to incorporate prior knowledge from manual tracing.

Previous work on sulcal detection mostly focused on detecting each sulcus separately. Curvature features were first used to develop semi-automated algorithms [6]–[8] with user specification of start/end points. Depth features with respect to a shrink wrap surface were also used to study sulcal regions on cortical surfaces [9] or find their line representations [10]. Based on the idea of skeletons [11], [12] and digital topology [13]–[15], medial models, or sulcal ribbons, of sulcal regions were constructed from volume images [16]–[21], but user inputs are still required to label specific sulcus from these results.

To automate the sulcal detection process, prior models were introduced to alleviate the difficulty of the problem. The principal component analysis (PCA) of point sets [22] was used to model the centroids of sulcal basins and help with the labeling [23]. Based on spherical maps of cortical surfaces, a hierarchical contour evolution scheme was developed using a PCA model of major sulci [24]. Graphical models were constructed with neural networks in [25] for simple surfaces, which are subsets of major sulci, computed with the skeletonization algorithm in [16], and then annealing techniques were used to label them. Based on a learning technique called *probabilistic boosting trees* [26], an automated approach was proposed in [27] to detect sulci from volume images, but each curve was treated separately.

In this work, we propose a joint detection approach that realizes sulcal detection via inference over graphical models of major sulci. We assume that each major sulcus is represented as a continuous curve on the cortical surface following a manual tracing protocol [28]. While this assumption may omit some interruptions over gyral regions, it is useful in improving the regularity when these curves are used to guide the mapping of cortical surfaces across population [3]. Based on boosting techniques, we not only incorporate the individual shape prior of each sulcal curve, but also model joint shape priors between neighboring sulci and integrate this information through belief propagation. From the practice of manual annotation, the use of pairwise shape priors seems a natural idea. For example, the pre-central sulcus usually needs to cross a gyrus to ensure it follows a route as parallel as possible to the central sulcus. In fact, such *local* dependencies are utilized fairly commonly to handle complicated situations in protocols for manual tracing [28].

Copyright (c) 2007 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

This work was funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 RR021813 entitled Center for Computational Biology (CCB). Information on the National Centers for Biomedical Computing can be obtained from <http://nihroadmap.nih.gov/bioinformatics>.

Y. Shi, Z. Tu, R. A. Dutton, A. D. Lee, I. Dinov, and P. M. Thompson are with the Laboratory of Neuro Imaging, Department of Neurology, UCLA School of Medicine, Los Angeles, CA 90095 USA.

A. L. Reiss is with the School of Medicine, Stanford University, Stanford, CA 94305 USA

A. M. Galaburda is with Harvard Medical School, Boston, MA 02215, USA

*Arthur W. Toga is with the Laboratory of Neuro Imaging, Department of Neurology, UCLA School of Medicine, Los Angeles, CA 90095 USA (toga@loni.ucla.edu).

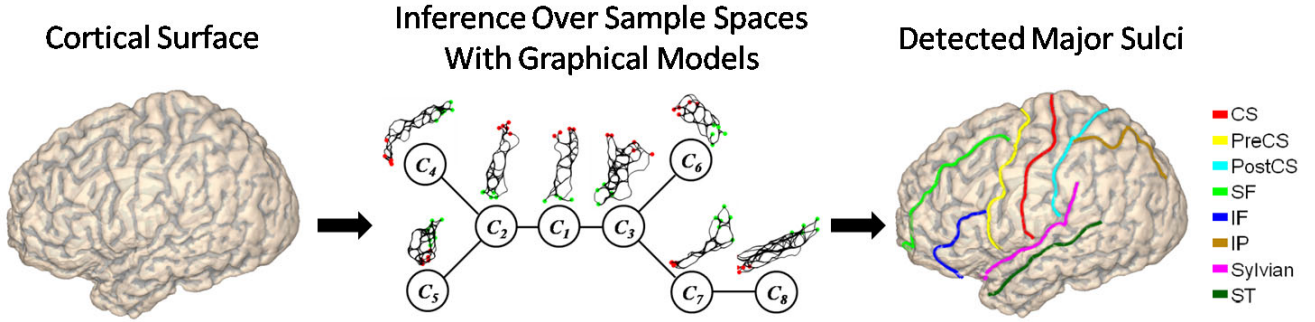


Fig. 1. An overview of our joint sulcal detection approach. The automatically detected sulci are plotted over the cortical surface with the color map on the right.

As an illustration, we provide an overview of our method in Fig. 1. In this example, the goal is to detect a set of eight major sulci on a cortical surface: the central sulcus (CS), pre-central sulcus (PreCS), post-central sulcus (PostCS), superior-frontal sulcus (SF), inferior-frontal sulcus (IF), intra-parietal sulcus (IP), sylvian fissure (Sylvian), and the superior-temporal sulcus (ST). An undirected graphical model of eight nodes is used to represent the Markovian relations of these sulci. Since the random variable at each node is a sulcal curve that lives in an infinitely dimensional shape space, it is generally difficult to perform inference directly over such spaces. We overcome this challenge by constructing a sample space containing a finite number of candidate curves, as plotted over each node in the graph in Fig. 1, greatly reducing the search range for each variable. To incorporate both individual and pairwise shape priors, we use boosting techniques to learn discriminative shape models and use them to define potential functions on the graphical model. Finally the max-product algorithm of belief propagation is used to find the MAP estimation from the joint sample spaces of the eight sulci as the sulcal detection results.

Compared with PCA models adopted in previous work [23], [24], the boosting approach we use does not need to impose the Gaussian assumption on shape models and can automatically select and fuse a large set of informative features to model both individual and pairwise shape priors. Our prior models are learned automatically from training data and there is no parameter to tune for different sulcal curves. Our method also works directly on cortical surfaces and does not need spherical maps of cortical surfaces [24].

Our work is most related to the sulci labeling algorithm proposed in [25], where the nodes of the graph are simple surfaces that oversample the major sulci and their labeling is realized by matching with a template graph learned from training data. In our work, we model each major sulcus as a continuous curve. In addition, the learning techniques and inference algorithms used in our work are different.

The rest of the paper is organized as follows. In section II, we first present the general framework for joint sulcal detection. After that, we develop the algorithm for generating sample spaces of candidate sulcal curves in section III. A learning-based approach for constructing potential functions of graphical models is proposed in section IV to model priors of sulcal curves. Experimental results are presented in section

V on a data set of 40 surfaces. Finally, we discuss possible future extensions in section VI.

II. THE JOINT DETECTION FRAMEWORK

In this section, we present our general framework for the joint detection of major sulci on cortical surfaces. By using a graphical model to represent Markovian relations of neighboring sulci, we realize automated sulcal detection by performing a MAP estimation over the sample spaces of sulcal lines.

Let \mathcal{M} denote the cortical surface and C_1, C_2, \dots, C_N be the set of major sulci to be detected on \mathcal{M} . To represent the Markovian relation among these sulci, we use an undirected graphical model $G = (V, E)$, where $V = \{C_1, C_2, \dots, C_N\}$ are the set of nodes, and E is the set of edges in the graph. As an example, a graphical model is shown in Fig. 1 for the eight major sulci: CS (C_1), PreCS (C_2), PostCS (C_3), SF (C_4), IF (C_5), IP (C_6), Sylvian (C_7), and ST (C_8). As the number of major sulci is typically small, we can construct such graph structures easily to encode desirable Markovian priors and it only needs to be done once for the same detection task.

Besides the graph structure, we need to specify the sample space for the random variable defined at each node and the potential functions to completely characterize the probabilistic graphical model. At each node in V , the random variable is a sulcal line and it can take values in a shape space of curves that is generally infinitely dimensional and difficult to analyze. One possible solution is to use dimension reduction techniques such as PCA models of curves [22] to generate each sulcal line as a linear combination of several basis functions. But the PCA models make the restrictive assumption of Gaussian distributions and there is no guarantee that the generated parametric curves will follow the sulcal regions. To overcome this problem, we develop a novel algorithm, which will be described in detail in section III, to automatically generate a set of candidate curves for each node by combining geometric features of sulcal regions and machine learning techniques. These curves are guaranteed to be on the cortical surface and they span a wide variety of possible routes for each sulcus of interest. With these candidate curves as the sample space S_i of each node C_i , we convert the sulcal detection problem to a tractable inference problem over a set of discrete random

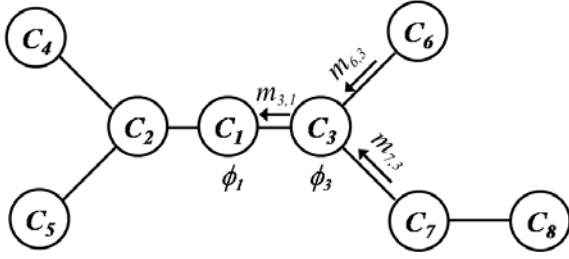


Fig. 2. An example of message passing in the graphical model of Fig. 1.

variables with the goal of selecting the best from the candidate curves.

Based on the sample spaces of candidate sulcal curves, we define two types of potential functions to complete the construction of the graphical model: the local evidence function $\phi_i : S_i \rightarrow \mathbb{R}$ at each node C_i and the compatibility function $\psi_{i,j} : S_i \times S_j \rightarrow \mathbb{R}$ for each edge $(C_i, C_j) \in E$. Given a candidate curve from S_i , the local evidence function ϕ_i gives the likelihood of this curve being the desirable sulcus to be detected. The compatibility function $\psi_{i,j}$ represents joint shape priors for two neighboring sulci (C_i, C_j) and measures how likely any two curves from $S_i \times S_j$ can co-exist as neighbors. To incorporate such individual and joint shape priors, we propose a discriminative approach based on AdaBoost [5] in section IV to learn both types of potential functions from manually annotated training data. With the discriminative approach, we can use a flexible and large set of features derived from training data and selectively combine them with boosting techniques to form the potential functions, so there is no need of specifying parametric forms for either the individual or joint shape prior models of sulcal lines.

The undirected graphical model defined above is a Markov random field, so the joint distribution of all the sulci can be factorized as a product of potential functions:

$$p(C_1, \dots, C_N) = \frac{1}{Z} \prod_{(C_i, C_j) \in E} \psi_{i,j}(C_i, C_j) \prod_{C_i \in V} \phi_i(C_i) \quad (1)$$

where Z is the partition function for normalization. The task of finding the optimal set of curves $(C_1^*, C_2^*, \dots, C_N^*)$ in the space $S_1 \times S_2 \times \dots \times S_N$ is then a MAP estimation problem defined as follows:

$$(C_1^*, C_2^*, \dots, C_N^*) = \underset{C_1, C_2, \dots, C_N}{\operatorname{argmax}} p(C_1, \dots, C_N). \quad (2)$$

To solve this inference problem over graphical models, we use the max-product algorithm of belief propagation [4], [29] because it can efficiently compute the optimal solution for tree-structured graphs and also demonstrated very good performance on graphs with cycles in various applications [30], [31]. With belief propagation, each node in the graph receives and sends out messages at every iteration of the algorithm. For a node C_i , the message it sends to its neighbor C_j is defined as:

$$m_{i,j}(C_j) = \max_{C_i \in S_i} \psi_{i,j}(C_i, C_j) \phi_i(C_i) \prod_{C_k \in \mathcal{N}(C_i) \setminus C_j} m_{k,i}(C_i) \quad (3)$$

where $\mathcal{N}(C_i)$ are neighbors of C_i in the graph. This message takes into account not only the local evidence ϕ_i and the compatibility function $\psi_{i,j}$, but also the messages the node C_i received from its neighbors except C_j . As an illustration, we show in Fig. 2 the flow of messages from the node C_6 and C_7 to C_3 , and then to C_1 in the graphical model shown in Fig. 1. After the message passing procedure converges, we obtain the final belief at each node of the graph as:

$$b_i(C_i) = \phi_i(C_i) \prod_{C_j \in \mathcal{N}(C_i)} m_{j,i}(C_i) \quad i = 1, 2, \dots, N. \quad (4)$$

and also the pairwise belief of each edge as:

$$b_{i,j}(C_i, C_j) = \phi_i(C_i) \phi_j(C_j) \psi_{i,j}(C_i, C_j) \times \prod_{C_k \in \mathcal{N}(C_i) \setminus C_j} m_{k,i}(C_i) \prod_{C_k \in \mathcal{N}(C_j) \setminus C_i} m_{k,j}(C_j) \quad (5)$$

Based on the final beliefs, we find an optimal configuration of major sulci with the following procedure [32], [33].

1. Start from a node C_i and pick the optimal sulcus C_i^* at this node as the one that maximizes $b_i(\cdot)$.
2. For each node C_i visited, if it has a neighbor C_j unvisited, find the optimal solution for C_j by maximizing the pairwise belief function $C_j^* = \operatorname{argmax}_{C_j} b_{i,j}(C_i^*, C_j)$. Repeat step 2 until all nodes are visited.

For tree-structured graphs, the above algorithm guarantees to find the globally optimal solution for the MAP estimation problem in (2). It is possible that more than one solution achieves global optimality for MAP estimation over graphs, but in our experience this does not happen in any of our sulci detection experiments. Nevertheless, we choose to fix the starting node as the one corresponding to the central sulcus in step 1 of the above procedure to remove the potential ambiguity that exists theoretically.

III. SAMPLE SPACE GENERATION

Given a cortical surface \mathcal{M} represented as a genus zero triangular mesh, which we assume is a left hemispheric surface aligned in a standard ICBM space [34] with a 9-parameter affine registration including independent scaling in x-, y-, and z- directions to account for brain size differences, there are four main steps in our algorithm to generate a sample space for each node in the graphical model: 1) extract the skeleton of the sulcal regions; 2) partition the surface into the lateral and medial part; 3) compute a set of possible start/end points and route-control segments of candidate curves with a learning-based approach; 4) generate candidate curves via random walks on a graph built from the start/end points and route-control segments.

A. Sulcal skeleton extraction

As a first stage toward sample space generation, we use the algorithm of computing Hamilton-Jacobi skeletons on cortical surfaces [35] to extract the skeleton of sulcal regions on \mathcal{M} . For completeness, we briefly describe the main steps of computing the sulcal skeletons as illustrated in Fig. 3. Using the mean curvature of the cortical surface \mathcal{M} , it is

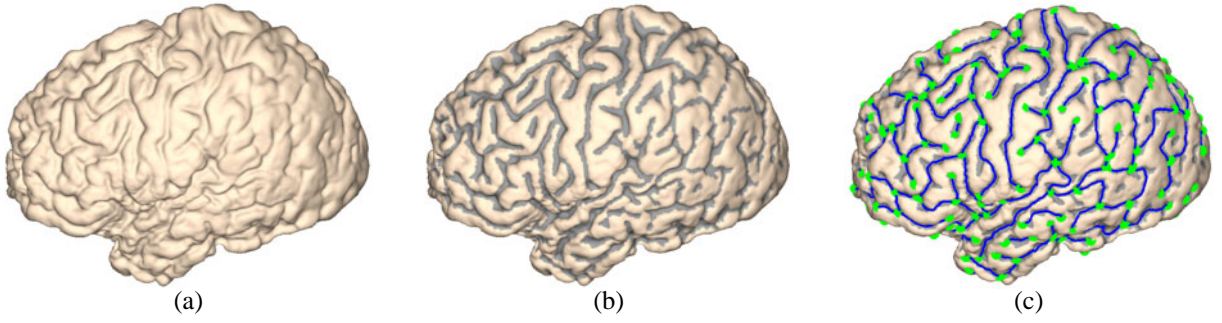


Fig. 3. Main steps in computing the sulcal skeletons of a cortical surface. (a) The cortical surface. (b) The partition of the surface into sulcal and gyral regions. (c) The Hamilton-Jacobi skeleton of the sulcal regions. End points of each branch are marked as green dots.

first partitioned into sulcal and gyral regions using graph cuts [36], [37] and the result is shown in Fig. 3(b). After that, the Hamilton-Jacobi skeleton method [38] is extended to triangular meshes to compute the skeleton of sulcal regions. A pruning process is finally applied to eliminate small branches with length below a specific threshold, which is 10mm in all our experiments. The sulcal skeletons are decomposed into a set of branches as shown in Fig. 3(c), where we have plotted the main body of the branches in blue and the end points in green. From the results we can see these sulcal skeletons capture the major folding patterns fairly well and provide a compact summary of cortex geometry.

B. Lateral/medial side partition

In the second stage we partition the hemispheric cortical surface \mathcal{M} into lateral and medial parts with a graph-cut algorithm. The resulting boundary between the lateral and medial side is then used to compute a set of features with the aim of providing intrinsic information about sulcal features in addition to absolute coordinates (*in millimeter*) in the ICBM space and improving the robustness to pose variations.

Before we apply the graph-cut algorithm, we first find a set of seed points for both the lateral and medial side. Since \mathcal{M} is a left hemispheric surface in the ICBM space, where the x-coordinate increases from left to right, we find a set of seed points X_l for the lateral side as vertices visible from the left side, i.e., the “x” direction, using the Z-buffer algorithm for visible surface determination in computer graphics. Similarly, the set of seed points X_m for the medial side are determined as vertices visible from the right side, i.e., the “-x” direction.

Because there are hidden regions invisible from either the left or right side, the two sets X_l and X_m do not form a complete partition of the surface. To achieve this goal, we minimize the following energy function to separate \mathcal{M} into the lateral side R_l and the medial side R_m :

$$E(R_l, R_m) = \sum_{\mathcal{V}_i \in R_l} d_{\mathcal{M}}(X_l, \mathcal{V}_i) + \sum_{\mathcal{V}_i \in R_m} d_{\mathcal{M}}(X_m, \mathcal{V}_i) + \lambda \sum_{i=1}^K \sum_{\mathcal{V}_j \in \mathcal{N}(\mathcal{V}_i)} \delta(\mathcal{V}_i, \mathcal{V}_j) \quad (6)$$

where \mathcal{V}_i and \mathcal{V}_j are vertices on \mathcal{M} , K is the total number of vertices, $d_{\mathcal{M}}(\cdot, \cdot)$ denote the geodesic distance between two point sets on \mathcal{M} that can be computed numerically with the

fast marching algorithm on triangular meshes [39], and the delta function δ is defined as one when \mathcal{V}_i and \mathcal{V}_j , a vertex in its one-ring neighborhood $\mathcal{N}(\mathcal{V}_i)$, belong to different regions and zero otherwise. The first two energy terms require R_l and R_m to be close to their seed points, the third energy term provides regularization for boundary smoothness and the non-negative parameter λ controls the weight of regularization. To minimize the energy, the same graph-cut algorithm used in stage one for partitioning \mathcal{M} into sulcal and gyral regions is applied to find the solution. Since this is a binary optimization problem, the graph-cut technique ensures the global optimality of the separation result [36], [37].

As an example, we show in Fig. 4 the partition results for the surface in Fig. 3(a). Choosing a proper regularization parameter ensures there will be no holes in R_l and R_m and their boundary is a simple curve. In our experience, the parameter $\lambda = 10$ gives very robust performance. With this parameter, our algorithm is able to successfully partition all of the 40 cortical surfaces used our experiments into only two connected components corresponding to the lateral and medial parts.

Once we have the partition results R_l and R_m , we find three boundary vertices, shown as red dots in Fig. 4, that have the largest y -coordinate, the smallest y -coordinate, and the smallest z -coordinate, respectively, and use them to divide the boundary between R_l and R_m into three curves BC_1, BC_2, BC_3 shown as the green, blue and cyan curve in Fig. 4 (a) and (b). We have also plotted the three curves from all 40 surfaces used in our experiments in Fig. 4(c). We can see these curves are clustered fairly closely in the ICBM space and this helps demonstrate the robustness of our partition algorithm. Using these three curves, we can compute the *landmark context* feature [40] defined at each vertex as $LC = [d_{BC_1}, d_{BC_2}, d_{BC_3}]$ to provide an intrinsic characterization of locations on \mathcal{M} , where d_{BC_j} ($j = 1, 2, 3$) is the geodesic distance to the curve BC_j . While the landmark context feature is not necessarily unique over the surface, it provides very intuitive characterizations of the intrinsic locations of major sulci on the cortical surface using distances to the three curves. For example, the distance to the curve BC_1 is useful in describing the almost parallel path to the medial wall the SF sulcus usually takes. This distance is also useful to describe the medial-to-lateral trend of the CS, PreCS, and PostCS. The distance to the curve BC_2 is valuable in

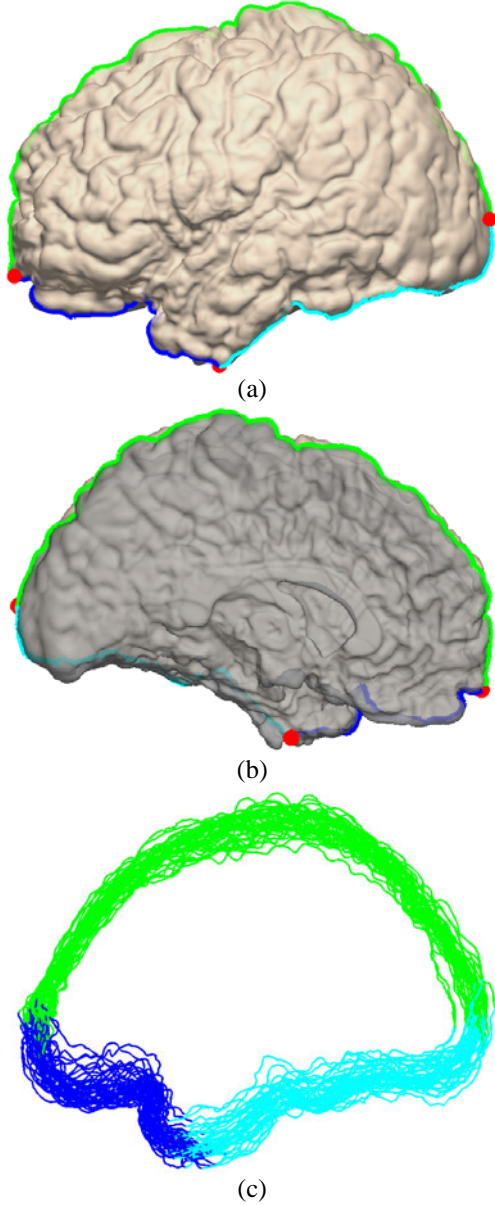


Fig. 4. The result of partitioning the cortical surface in Fig. 3(a) into the lateral(bright) and medial(dark) side. (a) The lateral view. (b) The medial view. The boundary between the two regions is divided into three curves: BC_1 (green), BC_2 (blue), and BC_3 (cyan). (c) The landmark curves from 40 cortical surfaces.

characterizing the intrinsic location of the frontal part of the SF, IF, ST sulcus and the sylvian fissure. With the distance to the curve BC_3 , we can quantify effectively the almost parallel relation between the ST sulcus and the sylvian fissure. In the next stage, we will use both the ICBM coordinate and the landmark context feature to characterize relative locations on \mathcal{M} for the learning algorithms.

C. Compute start/end points and route-control segments

By computing the skeleton of sulcal regions, we greatly reduce the search range for the start point X_i^s and end point X_i^e of a sulcal curve C_i . Let us denote the set of end points for the branches of sulcal skeletons, i.e., the green dots in Fig.

TABLE I
THE ADABOOST ALGORITHM [5].

Given training data: $(x_1, y_1, w_1^1), \dots, (x_n, y_n, w_n^1)$ where x_i are the sample data, $y_i \in \{-1, 1\}$ are the corresponding class labels, and w_i^1 are the initial weights.

For $t = 1, \dots, T$

- Train a weak classifier h_t given the current weights.
- Compute the weighted error rate ϵ_t of the classifier h_t .
- Update the weights:

$$w_i^{t+1} = \frac{w_i^t e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

where $\alpha_t = \log((1 - \epsilon_t)/\epsilon_t)/2$ and Z_t is a normalization constant such that $\sum_{i=1}^n w_i^{t+1} = 1$.

Output the final classifier $H = \text{sign}(f)$ with the decision function $f = \sum_{t=1}^T \alpha_t h_t$.

3(c), as \mathcal{A} . For each branch of the skeleton, we also divide it into $\lfloor \mathcal{L}/10 \rfloor$ segments of equal length around 10mm, where \mathcal{L} is the length of the branch and $\lfloor x \rfloor$ represents the largest integer less than or equal to x . We denote the set of such segments from all the branches as \mathcal{B} . In this third stage of our sample space generation algorithm, we use classifiers learned from training data to pick out a set of candidate points from the set \mathcal{A} for X_i^s and X_i^e , and a set of segments from \mathcal{B} , which we denote as *route-control* segments, that help better control the intermediate path the sample curves should follow from X_i^s to X_i^e .

To train these classifiers, we derive training data from a set of \mathcal{P} cortical surfaces with manually traced sulci and use AdaBoost [5] as our learning algorithm because it is easy to implement, flexible to incorporate various features, and robust to over fitting. As listed in Table I, the main idea of boosting is to form a strong classifier by combining a series of weak classifiers with their weights chosen adaptively based on their classification performance. In order to train the classifier for the start/end point of a sulcal curve C_i , we form the training data as follows by using the start point of C_i as an example. For each of the \mathcal{P} cortical surfaces in the training data, we compute the set \mathcal{A} of end points of sulcal skeletons. On each surface there is a manually labeled start point for the sulcal curve C_i and we compute the distance between this point and all points in \mathcal{A} . For each point in \mathcal{A} , if this distance is less than 5mm, we assign the label +1. For all other points in \mathcal{A} , we assign a label -1. Combining all the results from the \mathcal{P} surfaces, we form the training data for the start point of the sulcal curves C_i . The training data for its end point can be formed similarly. The features we use in our learning algorithm include the ICBM coordinate, the landmark context, and their individual components. For 1D features, we learn a decision stump as the weak classifier. For 3D features, we learn a perceptron using the pocket algorithm with ratchet [41] as the weak classifier. Both the decision stump and perceptron are linear classifiers in the form:

$$h(x) = \begin{cases} 1 & \text{if } \mu^T x + \xi > 0; \\ -1 & \text{else} \end{cases} \quad (7)$$

where x is the feature in \mathbb{R}^1 for the decision stump, and \mathbb{R}^3 for the perceptron. For the decision stump, (μ, ξ) are coefficients

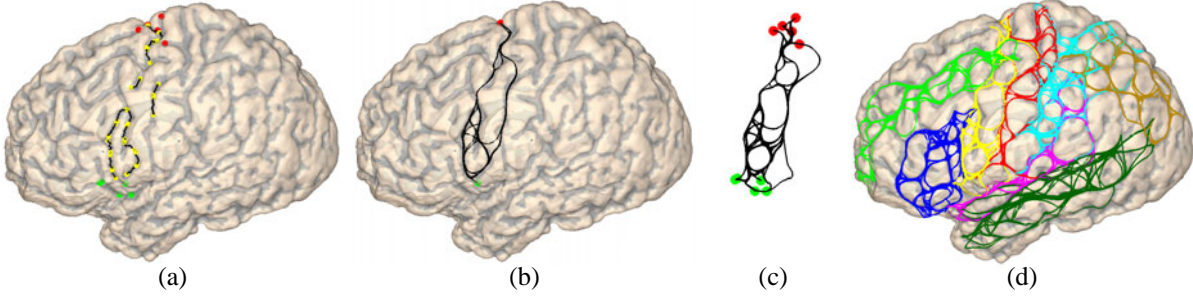


Fig. 5. The process of sample space generation. (a) The candidate start (red) and end (green) points of the precentral sulcus, and its route-control segments whose both ends are marked as yellow dots. (b) The sample curves generated with 30 random walks between a pair of start and end points of the precentral sulcus. (c) The sample space of precentral sulcus as composed of curves connecting all the start and end points. (d) The sample spaces of all eight major sulci in the graphical model of Fig. 1 are plotted with the same color map in Fig. 1.

of a 1D linear classifier. In the case of the perceptron, they are coefficients of a 3D linear classifier. While the perceptron seems sufficient for 3D features such as the ICBM coordinates and landmark context, we have chosen to also include the decision stump for the individual components of these 3D features because the training data is noisy and the 3D classifier may not be as specific as the 1D classifier in learning information that can be better captured by 1D features. So the decision stump can be viewed as a robust version of the perceptron useful to learn explicitly 1D information such as the distance to the medial wall. The AdaBoost algorithm is then used to selectively combine information from these features and form the final classifier.

For each sulcus C_i , we also use AdaBoost to train a classifier to pick out a set of route-control segments from the set \mathcal{B} that should be on the sample curves. For each of the P cortical surface in the training data, we first compute the set \mathcal{B} of skeleton segments. For every segment in \mathcal{B} , we then compute its Hausdorff distance to the manually traced curve on this surface. If the Hausdorff distance is less than $5mm$, we assign a label $+1$ to this segment; otherwise, a label -1 is assigned. Repeating the above procedure for all the P cortical surfaces, we form the training data for the route-control segments of C_i . For each segment, we compute the mean and difference of the ICBM coordinates, the landmark context features, and their individual components at its two end points as the features used for classification. The same learning algorithm described above is then used to learn the classifier for route-control segments.

D. Candidate curve generation via random walks

In this fourth stage, we generate sample curves based on random walks over a directed graph constructed from the candidate start/end points and route-control segments for a major sulcus C_i . Given a pair of candidate start point X_i^s and end point X_i^e , we order all the route-control segments of C_i according to their geodesic distance to X_i^s , which we denote as $RC = \{Q_1, Q_2, \dots, Q_J\}$ and a segment Q_{j1} is closer to X_i^s than Q_{j2} if $j1 < j2$. Similarly, we also order the two end points of a segment Q_j and denote them as Q_j^s and Q_j^e such that $d_{\mathcal{M}}(X_i^s, Q_j^s) \leq d_{\mathcal{M}}(X_i^s, Q_j^e)$, where $d_{\mathcal{M}}(\cdot, \cdot)$ denotes the geodesic distance between two points on \mathcal{M} as in (6).

The directed attributed graph $\vec{G} = (\vec{V}, \vec{E})$ for generating sample curves from X_i^s to X_i^e is composed of a set of nodes $\vec{V} = \{X_i^s, Q_1, Q_2, \dots, Q_J, X_i^e\}$ and a set of direct edges $\vec{E} = \vec{E}_1 \cup \vec{E}_2 \cup \vec{E}_3$. The set $\vec{E}_1 = \{X_i^s Q_j | j = 1, 2, \dots, J\}$ are directed edges from X_i^s to all the segments in RC and their weights are defined as $1/d_{\mathcal{M}}(X_i^s, Q_j^s)$. The set $\vec{E}_2 = \{Q_j X_i^e | j = 1, 2, \dots, J\}$ are directed edges from all the segments in RC to X_i^e and their weights are defined as $1/d_{\mathcal{M}}(Q_j^e, X_i^e)$. The set $\vec{E}_3 = \{Q_{j1} Q_{j2} | \text{if } d_{\mathcal{M}}(X_i^s, Q_{j1}^e) < d_{\mathcal{M}}(X_i^s, Q_{j2}^s), 1 \leq j1, j2 \leq J\}$ are composed of directed edges $Q_{j1} Q_{j2}$ between segments in RC . To ensure there are no loops in the graph, we require the end point of Q_{j1} to be closer to X_i^s than the start point of Q_{j2} . The weights for edges in \vec{E}_3 are defined as $1/d_{\mathcal{M}}(Q_{j1}^e, Q_{j2}^s)$.

To generate a sample curve, we perform a random walk in \vec{G} to find a path from X_i^s to X_i^e . Starting from the node X_i^s , we pick the next node by randomly choosing an edge from all the direct edges starting from X_i^s with a probability in proportion to the weights of these edges. The process is repeated until we reach the node X_i^e . Because we have defined the weights of edges inversely proportional to the geodesic distance between neighboring nodes, edges connecting closer nodes will have a higher chance of being visited in the random walk. Since there are no loops in \vec{G} and all the router-control segments are connected to X_i^s and X_i^e , any random walk starting from X_i^s is guaranteed to stop at X_i^e . By repeating the random walk multiple times, we can generate sample curves covering various routes from the start point to the end point. This is important as the major sulci are not necessarily the shortest path from X_i^s to X_i^e .

Suppose the path of the random walk in the graph \vec{G} is $X_i^s \rightarrow Q_{j1} \rightarrow Q_{j2} \rightarrow \dots \rightarrow Q_{jn} \rightarrow X_i^e$, we generate the sample curve by connecting a series of curve segments: $\overline{(X_i^s, Q_{j1}^s)} \rightarrow \overline{(Q_{j1}^e, Q_{j2}^s)} \rightarrow Q_{j2} \rightarrow \dots \rightarrow Q_{jn} \rightarrow \overline{(Q_{jn}^e, X_i^e)}$, where $\overline{(\cdot, \cdot)}$ represents a weighted geodesic path connecting two points on \mathcal{M} . Because it is possible to jump from a curve segment to a relatively far away curve segment during the random walk, we need to design the weighted geodesics to ensure the path connecting them passes through sulcal regions whenever possible. Thus we define the weight

function for computing the geodesic as

$$F = e^{-D} \quad (8)$$

where D is the distance transform of the sulcal skeletons, so points closer to the skeleton will have higher speeds. To find each path numerically, we use the fast marching algorithm on triangular meshes [39] to solve the Eikonal equation on \mathcal{M} :

$$\nabla d^w F = 1 \quad (9)$$

and trace backward along the gradient direction of d^w to find the geodesic path.

As an example, we show in Fig. 5(b) the sample curves between a start point (the red dot) and end point (the green dot) generated with random walks on the graph constructed with the route-control segments shown in Fig. 5(a). For each pair of start/end points, we typically generate 30 candidate curves in our experiments. In this case, there are 5 start and 5 end points, so we obtain a sample space of 750 candidate curves as shown in Fig. 5(c). Similarly, we can generate the sample space for other major sulci of interest. The sample spaces of the eight sulci in the graphical model of Fig. 1 are plotted in Fig. 5(d) with the color map in Fig. 1. In this case, the usefulness of the random walking process can be best illustrated in the sample space generated for the IF sulcus as the shortest path is clearly not the most desirable. It is clear that most of the sample curves shown in Fig. 5(d) are not neuro-biologically valid sulci. There are also overlaps between candidate curves of different sulci as it is possible for some curve segments being classified as route-control segments by multiple sulci. To ensure that the belief propagation algorithm can handle these cases correctly, we learn the potential functions of the graphical model from training data to incorporate prior knowledge about individual and neighboring sulci.

IV. LEARNING POTENTIAL FUNCTIONS

In this section, we describe our learning-based approach to construct both the local evidence functions ϕ_i and compatibility functions $\psi_{i,j}$ over the sample spaces of sulcal curves. For each potential function, we compute a large set of features and let the boosting technique automatically pick out the most informative features to model the individual and pairwise shape priors. Using the classifier learned by AdaBoost, we then define the potential function based on its connection to logistic regression [42].

For both the local evidence functions and compatibility functions, we assume a training data set of \mathcal{P} cortical surfaces in the ICBM space with manually labeled sulcal lines. For each surface, we compute the sample space S_i for each node C_i in the graphical model.

A. Local evidence functions

To learn the local evidence function ϕ_i for a sulcus C_i , we form its training data as follows. We assign all the manually traced curves on the \mathcal{P} surfaces the label +1. For a curve from S_i of each surface, we assign it the label -1 if more than 50%

of the points on the curve have a minimum distance of 10mm to the corresponding manually traced sulcus.

To characterize these curves, we use the Haar wavelet transform to compute a set of multi-scale features. More specifically, we uniformly sample each curve into $L_{Haar} = 32$ points. The Haar wavelet transform is then computed for the ICBM coordinates, landmark context features, and their individual components defined at these uniformly sample points. As a result, we have $2L_{Haar}$ 3D features from the ICBM coordinates and landmark context features, and $6L_{Haar}$ 1D features from their individual components. This large set of features provides a multi-scale description of the location and orientation of the curves.

Using these features, the AdaBoost algorithm combines a series of weak classifiers to form a final decision function for a curve $C_i \in S_i$:

$$f(C_i) = \sum_{t=1}^T \alpha_t h_t(C_i) \quad (10)$$

where h_t is the t -th weak classifier, which is a decision stump for 1D features and perceptron for 3D features, α_t is the weight for this classifier, and T is the total number of weak classifiers. It is shown in [42] that AdaBoost approximates logistic regression and the learned decision function can be used to estimate the probability of a class label, thus we follow [42] and define the local evidence function as

$$\phi_i(C_i) = \frac{e^{2f(C_i)}}{1 + e^{2f(C_i)}} \quad \forall C_i \in S_i. \quad (11)$$

The range of the local evidence function is between (0,1) and it approaches 1 when $f(C_i)$ is large for a curve $C_i \in S_i$, which suggests this curve bears strong similarity to manually traced sulcal lines in the training data. On the other hand, it approaches zero for curves with negative decision function values.

B. Compatibility functions

For two nodes C_i and C_j in the graphical model G , we follow a similar process to learn their compatibility function $\psi_{i,j}$, but with a different set of features to capture their joint shape priors. Given a cortical surface, we generate the sample spaces S_i and S_j for these two sulci, and the value $\psi_{i,j}(C_i, C_j)$ measures how compatible a pair of curves $(C_i, C_j) \in S_i \times S_j$ being the two major sulci. The training data to learn $\psi_{i,j}$ thus are also composed of curve pairs (C_i, C_j) that we form as follows. For the \mathcal{P} pairs of manually traced sulci for C_i and C_j on the \mathcal{P} surfaces in the training data, we assign a label +1. For each of the \mathcal{P} surface, we compute the sample space S_i and S_j and assign a label -1 for the set of curve pairs (C_i, C_j) generated by associating each curve $C_i \in S_i$ with a randomly picked curve $C_j \in S_j$ with the goal of representing possible cases of incompatible curves. By repeating the above procedure for each edge in the graphical model G , we can generate different training data for other neighboring sulci.

As inputs to the weak classifiers used in AdaBoost, we design a set of multi-scale features to model the joint configuration for each pair of curves (C_i, C_j) . Let Υ denote the

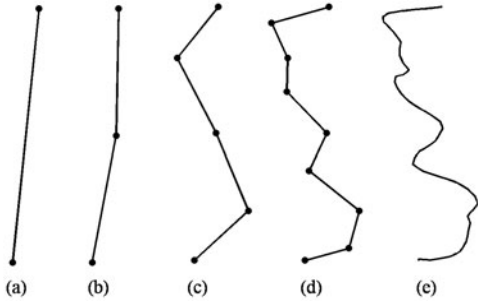


Fig. 6. The multilevel decomposition of a curve. (a) $\eta = 0$. (b) $\eta = 1$. (c) $\eta = 2$. (d) $\eta = 3$. (e) The original curve.

maximum number of levels we want to compute the features. We re-sample each curve C_i and C_j into $2^\Upsilon + 1$ equally spaced points. Let $p_0, p_1, \dots, p_{2^\Upsilon}$ be the $2^\Upsilon + 1$ points on C_i , we then approximate it with a set of 2^η straight line segments at the level $0 \leq \eta \leq \Upsilon$:

$$MC(\eta) = \{\overline{p_{k2^{\Upsilon-\eta}} p_{(k+1)2^{\Upsilon-\eta}}} | 0 \leq k < 2^\eta\}, \quad (12)$$

where $\overline{p_{k2^{\Upsilon-\eta}} p_{(k+1)2^{\Upsilon-\eta}}}$ denote the line segment connecting the two points $p_{k2^{\Upsilon-\eta}}$ and $p_{(k+1)2^{\Upsilon-\eta}}$. As shown in Fig. 6, line segments at the coarse scale captures the global trend of each curve, while the line segments at finer scales provide more local information. Similarly, the curve C_j is also decomposed into the same number of levels and we use the relation between line segments from these two curves to characterize their configuration at each level. More specifically, for each line segment LS_1 of C_i and LS_2 of C_j at a level η , we compute the angle between them and the shortest displacement vector from points on LS_1 to LS_2 . Repeat this procedure for all levels, we obtain a set of multi-scale features to describe the relative position of curves. The AdaBoost algorithm is then used to learn a strong classifier for the curve pair (C_i, C_j) . Similar to the definition of local evidence functions, we define the compatibility function between C_i and C_j as:

$$\psi_{i,j}(C_i, C_j) = \frac{e^{2f(C_i, C_j)}}{1 + e^{2f(C_i, C_j)}} \quad \forall (C_i, C_j) \in S_i \times S_j. \quad (13)$$

where $f(\cdot, \cdot)$ is the decision function learned with AdaBoost using features described above.

Applying the above learning algorithms to each node and every pair of neighboring sulci in a graphical model, we can learn all the local evidence and compatibility functions, which complete the construction of the graphical model. After that, the belief propagation algorithm can be applied to detect sulci jointly on cortical surfaces from the sample spaces of major sulci.

V. EXPERIMENTAL RESULTS

In this section, we present experimental results to demonstrate our joint sulcal detection algorithm on a data set of 40 left hemispheric surfaces. These surfaces represent the boundary between the gray matter and cerebrospinal fluid (CSF), and were generated from MRI images in the ICBM space using a surface extraction algorithm [43] and they all have genus zero topology. While these surfaces may not

capture the deepest parts of sulcal regions, the regularity they share makes it easier to compare across population and perform group studies. More details on the MRI imaging and post-processing protocols can be found in [44]. A set of eight major sulci were manually labeled on each surface for training and validation, which include the central sulcus (CS), pre-central sulcus (PreCS), post-central sulcus (PostCS), superior-frontal sulcus (SF), inferior-frontal sulcus (IF), intra-parietal sulcus (IP), sylvian fissure (Sylvian), and the superior-temporal sulcus (ST).

In our experiments, we use the graphical model in Fig. 1 to demonstrate the joint detection method. As a common practice in graph-based estimation, belief propagation is also often applied to graphs with loops. This may allow us to incorporate more neighboring priors between sulcal curves, but no theoretical guarantee of global optimality exists for graphs with loops. So there is a tradeoff between using more complicated models and the tractability in obtaining the optimal solution. We have chosen to use tree-structured graphical models in our experiments as this allows us to focus more on the construction of sample spaces of candidate curves and shape priors.

In the first experiment, we learn from a set of training data the classifiers used in sample space generation and the potential functions in the graphical model, and then demonstrate the sulcal detection algorithm on surfaces in the training data. After that we apply the joint detection algorithm to a set of testing data and validate its performance quantitatively. In our third experiment, we demonstrate the advantage of the joint detection approach by comparing with results obtained without using the graphical model.

A. Graphical model training

Among the 40 surfaces in our data we randomly pick 20 surfaces, together with their manually traced sulci, as the training data. As an illustration, we show three examples from the training data in Fig. 7 (a), (b), and (c). The other 20 surfaces are used as the testing data to evaluate the performance of the joint sulcal detection algorithm.

In this learning stage, we first compute the sulcal skeletons for all the surfaces in the training data. After that the landmark context features derived from the boundary of the lateral/medial partition are computed with the graph-cut approach developed in section III-B. Using results obtained in these two steps, the AdaBoost algorithm is applied to learn the classifiers for the start/end points and route-control segments of each sulcal curve as described in section III-C. On every surface, we then run the random-walk algorithm in section III-D for each sulcal curve, or node of the graphical model, to generate its sample space of candidate curves. For the surfaces in Fig. 7 (a), (b), and (c), the corresponding sample spaces of the eight major sulci are plotted in Fig. 7 (d), (e), and (f). Finally the potential functions in the graphical model are learned following the procedure in section IV.

As a first step to examine the graphical model we learned from training data, we perform MAP estimation on the sample spaces of each surface in training data using the belief

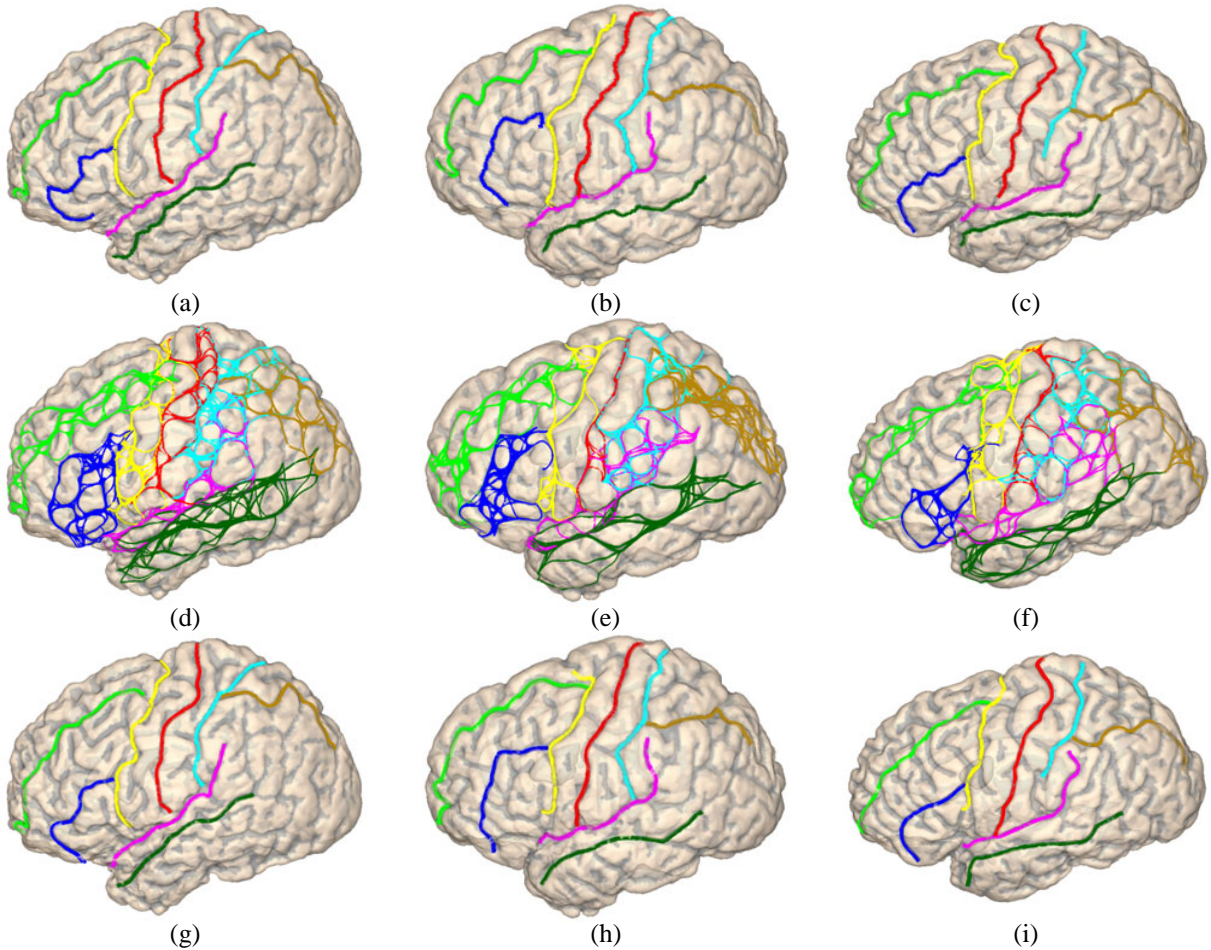


Fig. 7. Three surfaces in the training data. (a)(b)(c) Cortical surfaces with manually traced sulcal curves. (d)(e)(f) Sample spaces of the eight sulci. (g)(h)(i) Automatically detected sulcal curves plotted on surfaces. (All curves are plotted with the color map in Fig. 1.)

TABLE II

QUANTILE STATISTICS OF d_{am} AND d_{ma} FROM SULCI DETECTED IN TRAINING DATA. ALL DISTANCES ARE IN MILLIMETERS.

		CS	PreCS	PostCS	SF	IF	IP	Sylvian	ST
d_{am}	50th percentile	0.70	1.05	1.01	1.24	1.15	0.97	0.93	1.04
	70th percentile	0.98	1.91	1.86	2.25	3.28	2.01	1.58	1.94
	90th percentile	1.66	6.35	9.76	8.92	8.26	6.87	4.60	8.54
d_{ma}	50th percentile	0.75	1.18	1.14	1.38	1.31	1.12	1.05	1.26
	70th percentile	1.14	2.30	2.19	2.55	3.34	2.18	1.95	2.56
	90th percentile	2.09	8.15	12.75	9.29	8.62	7.35	6.62	11.21

propagation algorithm described in section II. By presenting results from the training data, we demonstrate the ability of our learning-based approach in capturing shape priors from manual tracing. These results will also be used to compare with results from the testing data to illustrate the generalization ability of our method. For the three examples in Fig. 7, we have plotted in Fig. 7 (g), (h), and (i) the detected sulci. Compared with the manually traced curves in Fig. 7 (a), (b), and (c), we can see the detected sulci travel mostly along the same routes through sulcal regions. The geometric relations of neighboring sulci, such as the junctions between the IF and PreCS, are also correctly followed because both angle and displacement features are used in modeling the compatibility functions of these neighboring sulci.

Besides the visual results, we next present more detailed

quantitative measures to compare automatically detected sulci with manual results. In previous works [19], [24], [27], statistics such as the mean and standard deviation of distances between points on detected sulci and manually traced curves were used. We extend these measures and use more detailed quantile statistics in our experiments. With quantile statistics, we can characterize how well the detected curves align with the main body of the major sulci at different levels, which maybe anatomically more reasonable considering the difficulty and the resulting variability in deciding the starting and ending parts of sulcal curves even for human tracers.

To compute the quantile statistics, we first calculate two distances between automatically detected curves and manual results. For each point on an automatically detected curve, we compute its minimum distance to the corresponding manually

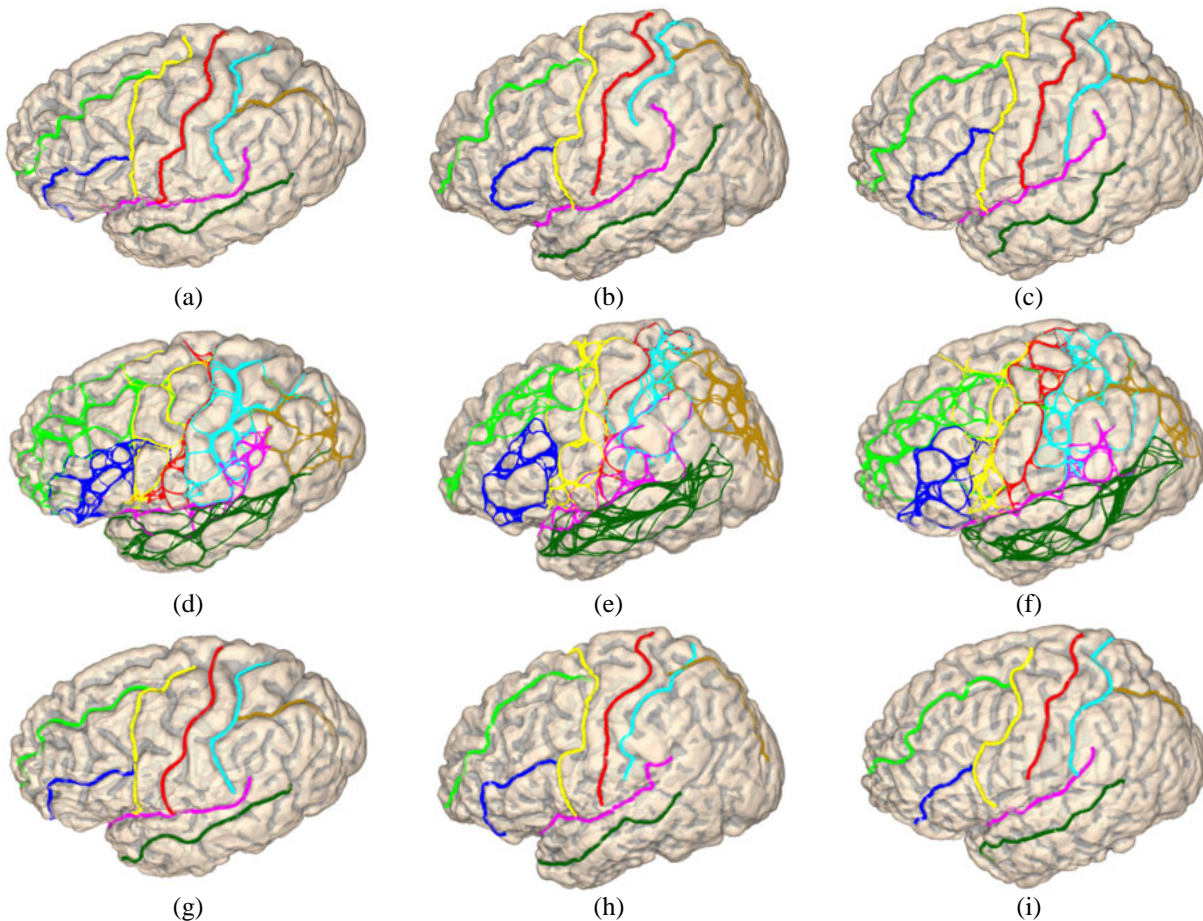


Fig. 8. Sulcal detection results on three cortical surfaces in the testing data. (a)(b)(c) Cortical surfaces with manually traced sulci. (d)(e)(f) Sample spaces of the eight sulci. (g)(h)(i) Automatically detected sulcal curves plotted on surfaces. (All curves are plotted with the color map in Fig. 1.)

TABLE III

QUANTILE STATISTICS OF d_{am} AND d_{ma} FROM SULCI DETECTED JOINTLY IN TESTING DATA. ALL DISTANCES ARE IN MILLIMETERS.

		CS	PreCS	PostCS	SF	IF	IP	Sylvian	ST
d_{am}	50th percentile	0.70	1.23	1.05	1.46	1.55	0.92	1.34	1.33
	70th percentile	0.97	2.42	2.11	3.25	3.99	1.74	2.84	2.75
	90th percentile	1.70	8.07	9.32	8.98	9.17	8.60	7.36	7.64
d_{ma}	50th percentile	0.74	1.44	1.23	1.63	1.97	1.12	1.77	1.66
	70th percentile	1.14	3.32	2.93	3.38	5.16	2.24	4.13	4.19
	90th percentile	2.38	9.06	10.77	9.44	11.50	11.05	11.02	10.75

traced curve and denote this kind of distance as d_{am} . For each point on the manually traced curve, we also compute its minimum distance to the automatically detected curve and denote this distance as d_{ma} . For each major sulcus, we use the 20 pairs of automatically detected and manually traced curves on all the surfaces in training data to compute the quantile statistics of d_{am} and d_{ma} at the 50th, 70th, and 90th percentile.

The results for all eight sulci are listed in Table II. Each distance value in the table represents a cut-point on the cumulative distribution function (CDF) of either d_{am} or d_{ma} for a certain sulcus. For example, the last number in the column of central sulcus means that 90% of points on the manually traced curve have a distance $d_{ma} \leq 2.09mm$ to the automatically detected curve. For all eight sulci, we can see the 50th percentile of both d_{am} and d_{ma} are less than $1.5mm$, and the 70th percentile are less than $3.5mm$. Even for highly

variable sulci such as PostCS and IF, the 90th percentile of both d_{am} and d_{ma} are around $10mm$. So the results in Table II show that very good alignment has been achieved between the main body of automatically detected curves and manually traced ones.

B. Validation with testing data

In our previous experiment, we demonstrated very good sulcal detection results in Fig. 7 and Table II. For practical purposes, however, it is more important to examine the generalization abilities of the algorithm, i.e., its performance on testing data. In this experiment, we apply our algorithm to the 20 surfaces in the testing data and evaluate the results quantitatively with manually labeled sulcal lines.

There are two main steps in applying our algorithm to testing data. In the first step, we generate a sample space of

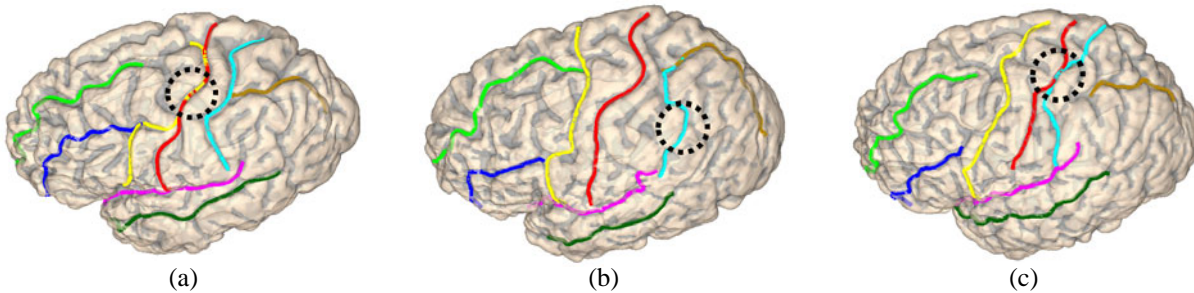


Fig. 9. Sulcal detection results without using graphical models. (All curves are plotted with the color map in Fig. 1.)

TABLE IV

QUANTILE STATISTICS OF d_{am} AND d_{ma} FROM SULCI DETECTED SEPARATELY IN TESTING DATA. ALL DISTANCES ARE IN MILLIMETERS.

		CS	PreCS	PostCS	SF	IF	IP	Sylvian	ST
d_{am}	50th percentile	0.69	1.23	1.56	1.71	2.34	0.88	1.33	1.34
	70th percentile	0.99	2.96	7.52	4.87	7.29	1.65	2.86	3.24
	90th percentile	2.04	10.32	13.68	11.13	11.56	10.59	9.15	8.63
d_{ma}	50th percentile	0.74	1.54	1.90	1.83	2.42	1.13	1.71	1.78
	70th percentile	1.15	4.01	9.02	4.57	8.46	2.29	4.27	4.87
	90th percentile	2.82	10.95	16.42	11.10	13.52	12.61	11.79	11.65

candidate curves for each sulcus on a cortical surface. During this stage, the classifier for start/end points and route-control segments learned from training data are used. After that, belief propagation is applied to pick out the best combination of sulcal curves from their sample spaces. Using the classifiers learned from training data, there are very few parameters to tune when we apply the two steps to the testing data. The only parameter we need to adjust is the number of sample curves to generate for each pair of start/end points of a sulcal curve at the first step. When we increase this number, we get larger sample spaces covering more routes a sulcal curve can follow, but it also increases the computational cost because more random walks need to be performed. In our experiments, we set this parameter to 30 curves as in the example shown in section III-D. This usually generates a sample space containing around 1000 curves for each major sulcus. No significant gains are observed if we further increase this parameter. As an illustration, we visualize the results from these two steps on three surfaces in the testing data as shown in Fig. 8(a), (b), and (c). The sample spaces generated for each of the eight major sulci are plotted on the surfaces in Fig. 8(d), (e), and (f) using the same color map in Fig. 1. The detected sulcal curves are plotted on the surfaces in Fig. 8(g), (h), and (i). By comparing the automatically detected sulci with manually traced curves in Fig. 8(a), (b), and (c), we can see the automatically detected curves overall capture the main body of the sulci and agree with manual results very well.

To measure quantitatively the performance of our sulcal detection algorithm on testing data, we compute the same statistics as in the first experiment. For each major sulcus, we use the 20 pairs of automatically detected and manually traced curves in testing data to calculate the quantile statistics of d_{am} and d_{ma} at the 50th, 70th, and 90th percentile. The results for the eight sulci are listed in Table III. For all eight sulci, the 50th percentile of both d_{am} and d_{ma} are less than $2mm$, and the 70th percentile are less than or around $5mm$. Besides

the central sulcus, the 90th percentile of d_{am} and d_{ma} are around $10mm$, which is slightly worse than the performance on training data and this is mainly due to the high variability in the starting and ending parts of sulcal curves. One good example in illustrating this difficulty is the superior end of the PostCS. Following the tracing protocol [28], the manual tracer is able to consistently pick the posterior route whenever there are more than one choices in determining this part of the PostCS such as the example in Fig. 7(a). Our automated approach, however, may sometimes get confused and follow a posterior route that actually jumps across a gyrus as in Fig. 8(h). Another example is the difficulty in capturing the frontal part of the IF sulcus that usually bends backward. This is mainly because that the IF sulcus does not necessarily follow a weighted geodesic that we use to generate candidate curves. These kinds of situations contribute to the largest errors in the quantile statistics and point out directions of future improvements.

Overall the results in Table III show that our algorithm generalizes very well on testing data and the detected curves are able to capture the main body of major sulci accurately, which is especially encouraging for those sulci (PostCS, IF) that are highly variable.

C. Comparison with sulci detected separately

In this experiment, we compare the joint detection results in section V-B with sulci detected separately without taking into account pairwise priors between neighboring sulci, which is realized by choosing the curve in each sample space that maximizes the associated local evidence function.

For the three surfaces in Fig. 8(a), (b), and (c), we plot the independently detected sulci in Fig. 9 (a), (b), and (c), respectively. To highlight the differences between the detection results in Fig. 8 and Fig. 9, we have annotated with a dotted circle to identify one place on each surface in Fig. 9 where the independently detected curve failed to locate the corresponding

sulcus accurately. On the contrary, these kinds of mistakes were avoided in the results shown in Fig. 8(g), (h), and (i) because pairwise priors are incorporated. This demonstrates the value of the compatibility functions in improving sulci detection with joint shape priors.

Following the quantitative evaluation procedure in section V-B, we also compare independently detected sulci with manually annotated sulcal curves on the 20 surfaces in our testing data by computing the same quantile statistics. The results are listed in Table IV. From the numbers in Table III and IV, we can clearly see the advantage of the joint detection approach as it performs better on 40 of the 48 distance measures. For the 6 measures that the independent detection approach generated better results, it outperforms the joint detection method only by a slight margin of less than 0.1mm. The performance gain with the use of the graphical model is especially significant for more variable sulci such as the PostCS and IF as demonstrated by their quantile statistics at the 70th percentile for both d_{am} and d_{ma} in Table III and IV.

VI. DISCUSSION AND CONCLUSIONS

In this paper, we proposed a joint detection framework for the automated labeling of major sulci on cortical surfaces. By generating sample spaces of candidate curves for the major sulci, we are able to convert sulcal detection into a tractable inference problem over discrete random variables. To capture both individual and joint shape priors of sulcal curves, we use graphical models in our framework to encode Markovian relations between neighboring sulci and learn the potential functions automatically with AdaBoost.

With the aim of providing stable landmark curves for the mapping of cortical surfaces across population, we represent each major sulcus as a continuous curve in our work, which is useful for the analysis of anatomical quantities defined on cortical surfaces such as gray matter densities. On the other hand, this assumption simplifies the interruptions of the sulci over gyral regions that exist naturally. So when the sulcal anatomy is the target of analysis, it might be beneficial to study the detailed configuration of sulcal regions directly.

In our experiments, we demonstrated the training of a graphical model and applied it to automatically detect a set of eight major sulci on hemispheric cortical surfaces. These sulci are on the lateral surface of the cortex, but our method is general and can also be applied to detect other major sulci on the medial surface such as the calcarine sulcus. For the detection of secondary sulci that may or may not be present, for example the secondary cingulate sulcus, however, we cannot apply our method directly. A model selection process might be necessary to first determine the proper graphical model to use and then apply the joint detection algorithm we develop here.

As noted in our experiments, there are still difficulties in accurately detecting sulcal lines that tend to bend backward. To address this problem, we will improve in the future work the sample space generation algorithm for these sulci to ensure their sample spaces contain valid candidate curves. For example, we can train an additional classifier for the IF sulcus

to detect a route-control segment corresponding to the most frontal part of the sulcal line and use it to capture the bending between the start and end points.

A large set of features derived from the ICBM coordinates and landmark context features have been combined with AdaBoost to model shape priors of sulcal curves in our current work. The ability of this approach in modeling joint shape priors was demonstrated via comparisons with results detected without using graphical models. An interesting direction of future research is to include a feature selection process [45], [46] in our algorithm as many features contain redundant information. This may improve the effectiveness of our model. For example, this process could make the compatibility function of the PostCS and IP sulcus more sensitive to the spatial configuration between the closest line segments in their multilevel decompositions and help eliminate artifacts such as the slight overlap of these two sulcal curves in Fig. 8(h).

We have chosen to use tree-structured graphical models in our experiments because the belief propagation algorithm can efficiently compute the globally optimal solution on such models. This is, however, at the expense of leaving out potentially useful neighboring priors. To incorporate more joint shape priors, we will study the use of graphical models with loops in our future work. The same learning process developed here can still be used to construct the sample spaces of candidate curves and potential functions on such models, but the belief propagation algorithm has to be used with caution as there is no guarantee of global optimality anymore. More sophisticated optimization strategies such as the tree-reweighted message passing algorithm [47] and the primal-dual graph cut algorithm [48] will be investigated for MAP estimation on these graphical models with loops.

REFERENCES

- [1] M. Ono, S. Kubik, and C. Abarnathey, *Atlas of the Cerebral Sulci*. Thieme Medical Publishers, 1990.
- [2] D. C. Van Essen, "A population-average, landmark- and surface-based (PALS) atlas of human cerebral cortex," *NeuroImage*, vol. 28, no. 3, pp. 635–62, 2005.
- [3] P. M. Thompson, K. M. Hayashi, E. R. Sowell, N. Gogtay, J. N. Giedd, J. L. Rapoport, G. I. de Zubicaray, A. L. Janke, S. E. Rose, J. Semple, D. M. Doddrell, Y. Wang, T. G. M. van Erp, T. D. Cannon, and A. W. Toga, "Mapping cortical change in alzheimers disease, brain development, and schizophrenia," *NeuroImage*, vol. 23, pp. S2–S18, 2004.
- [4] J. Perl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo: Morgan Kaufman, 1988.
- [5] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [6] N. Khaneja, M. Miller, and U. Grenander, "Dynamic programming generation of curves on brain surfaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 11, pp. 1260–1265, 1998.
- [7] A. Bartesaghi and G. Sapiro, "A system for the generation of curves on 3D brain images," *Human Brain Mapping*, vol. 14, pp. 1–15, 2001.
- [8] L. M. Lui, Y. Wang, T. F. Chan, and P. M. Thompson, "Automatic landmark and its application to the optimization of brain conformal mapping," in *Proc. CVPR*, vol. 2, 2006, pp. 1784–1792.
- [9] M. E. Rettmann, X. Han, C. Xu, and J. L. Prince, "Automated sulcal segmentation using watersheds on the cortical surface," *NeuroImage*, vol. 15, no. 2, pp. 329–244, 2002.
- [10] C. Kao, M. Hofer, G. Sapiro, J. Stern, K. Rehm, and D. Rotterberg, "A geometric method for automatic extraction of sulcal fundi," *IEEE Trans. Med. Imag.*, vol. 26, no. 4, pp. 530–540, 2007.

- [11] H. Blum and R. Nagel, "Shape description using weighted symmetric axis features," *Pattern Recognition*, vol. 10, no. 3, pp. 167–180, 1978.
- [12] S. M. Pizer, D. S. Fritsch, P. A. Yushkevich, V. E. Johnson, and E. L. Chaney, "Segmentation, registration, and measurement of shape variation via image object shape," *IEEE Trans. Med. Imag.*, vol. 18, no. 10, pp. 851–865, 1999.
- [13] T. Kong and A. Rosenfeld, "Digital topology: introduction and survey," *Comput. Vis. Graph. Image Process.*, vol. 48, no. 3, pp. 357–393, Dec 1989.
- [14] G. Bertrand and R. Malgouyres, "Some topological properties of surfaces in Z^3 ," *J. Math. Imaging. Vis.*, vol. 11, pp. 207–221, 1999.
- [15] X. Han, C. Xu, and J. Prince, "A topology preserving level set method for geometric deformable models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 6, pp. 755–768, Jun 2003.
- [16] J.-F. Mangin, V. Frouin, I. Bloch, J. Régis, and J. López-Krahe, "From 3D magnetic resonance images to structural representations of the cortex topography using topology preserving deformations," *Journal of Mathematical Imaging and Vision*, vol. 5, no. 4, pp. 297–318, 1995.
- [17] G. Lohmann, "Extracting line representations of sulcal and gyral patterns in MR images of the human brain," *IEEE Trans. Med. Imag.*, vol. 17, no. 6, pp. 1040–1048, 1998.
- [18] M. Vaillant and C. Davatzikos, "Finding parametric representations of the cortical sulci using an active contour model," *Med. Image Anal.*, vol. 1, no. 4, pp. 295–315, 1996.
- [19] G. Goualher, E. Procyk, D. Collins, R. Venugopal, C. Barillot, and A. Evans, "Automated extraction and variability analysis of sulcal neuroanatomy," *IEEE Trans. Med. Imag.*, vol. 18, no. 3, pp. 206–217, 1999.
- [20] Y. Zhou, P. M. Thompson, and A. W. Toga, "Extracting and representing the cortical sulci," *IEEE Computer Graphics and Applications*, vol. 19, no. 3, pp. 49–55, 1999.
- [21] X. Zeng, L. Staib, R. Schultz, H. Tagare, L. Win, and J. Duncan, "A new approach to 3D sulcal ribbon finding from MR images," in *Proc. MICCAI*, 1999, pp. 148–157.
- [22] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models—their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.
- [23] G. Lohmann and D. Cramon, "Automatic labelling of the human cortical surface using sulcal basins," *Med. Image Anal.*, vol. 4, pp. 179–188, 2000.
- [24] X. Tao, J. Prince, and C. Davatzikos, "Using a statistical shape model to extract sulcal curves on the outer cortex of the human brain," *IEEE Trans. Med. Imag.*, vol. 21, no. 5, pp. 513–524, 2002.
- [25] D. Rivière, J.-F. Mangin, D. Papadopoulos-Orfanos, J. Martinez, V. Frouin, and J. Régis, "Automatic recognition of cortical sulci of the human brain using a congregation of neural networks," *Med. Image Anal.*, vol. 6, pp. 77–92, 2002.
- [26] Z. Tu, "Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering," in *Proc. ICCV*, vol. 2, 2005, pp. 1589–1596.
- [27] Z. Tu, S. Zheng, A. Yuille, A. Reiss, R. A. Dutton, A. Lee, A. Galaburda, I. Dinov, P. Thompson, and A. Toga, "Automated extraction of the cortical sulci based on a supervised learning approach," *IEEE Trans. Med. Imag.*, vol. 26, pp. 541–552, 2007.
- [28] [Online]. Available: http://www.loni.ucla.edu/~esowell/elevel/new_sulcvar.html
- [29] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Info. Theory*, vol. 46, pp. 325–343, 2000.
- [30] S. M. Aji, G. Horn, R. J. McEliece, and M. Xu, "Iterative min-sum decoding of tail-biting codes," in *Proc. IEEE Information Theory Workshop*, 1998, pp. 68–69.
- [31] W. T. Freeman and E. Pasztor, "Learning to estimate scenes from images," in *Proc. Neural Information Processing Systems (NIPS)*, vol. 2, 1998, pp. 775–781.
- [32] A. P. Dawid, "Applications of a general propagation for probabilistic expert systems," *Statistics and Computing*, vol. 2, pp. 25–36, 1992.
- [33] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree consistency and bounds on the performance of the max-product algorithm and its generalizations," *Statistics and Computing*, pp. 143–166, 2004.
- [34] J. C. Mazziotta, A. W. Toga, A. C. Evans, P. T. F. and J. Lancaster, K. Zilles, R. P. Woods, T. Paus, G. Simpson, B. Pike, C. J. Holmes, D. L. Collins, P. M. Thompson, D. MacDonald, T. Schormann, K. Amunts, N. Palomero-Gallagher, L. Parsons, K. L. Narr, and N. Kabani, "A probabilistic atlas and reference system for the human brain: international consortium for brain mapping," *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, vol. 356, pp. 1293–1322, 2001.
- [35] Y. Shi, P. Thompson, I. Dinov, and A. Toga, "Hamilton-Jacobi skeleton on cortical surfaces," *IEEE Trans. Med. Imag.*, vol. 27, no. 5, pp. 664–673, 2008.
- [36] Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. ICCV*, vol. 1, 2001, pp. 105–112.
- [37] Y. Boykov and V. Kolmogorov, "An experimental comparison of Min-Cut/Max-Flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 9, pp. 1124–1137, Sept 2004.
- [38] K. Siddiqi, S. Bouix, A. Tannebaum, and S. Zuker, "Hamilton-Jacobi skeletons," *Int'l Journal of Computer Vision*, vol. 48, no. 3, pp. 215–231, 2002.
- [39] R. Kimmel and J. A. Sethian, "Computing geodesic paths on manifolds," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 15, pp. 8431–8435, 1998.
- [40] Y. Shi, P. M. Thompson, I. Dinov, S. Osher, and A. W. Toga, "Direct cortical mapping via solving partial differential equations on implicit surfaces," *Med. Image Anal.*, vol. 11, no. 3, pp. 207–223, 2007.
- [41] S. Gallant, "Perceptron-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 1, no. 2, pp. 179–191, 1990.
- [42] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.
- [43] D. MacDonald, "A method for identifying geometrically simple surfaces from three dimensional images," Ph.D. dissertation, McGill Univ., Canada, 1998.
- [44] P. M. Thompson, A. D. Lee, R. A. Dutton, J. A. Geaga, K. M. Hayashi, M. A. Eckert, U. Bellugi, A. M. Galaburda, J. R. Korenberg, D. L. Mills, A. W. Toga, and A. L. Reiss, "Abnormal cortical complexity and thickness profiles mapped in Williams syndrome," *J. Neuroscience*, vol. 25, no. 16, pp. 4146–4158, 2005.
- [45] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- [46] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [47] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "MAP estimation via agreement on trees: message-passing and linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3697–3717, 2005.
- [48] N. Komodakis and G. Tziritas, "Approximate labeling via graph cuts based on linear programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 8, pp. 1436–1453, 2007.