Joint Sulci Detection Using Graphical Models and Boosted Priors

Yonggang Shi[1], Zhuowen Tu[1], Allan L. Reiss[2], Rebecca A. Dutton[1], Agatha D. Lee[1], Albert M. Galaburda[3], Ivo Dinov[1], Paul M. Thompson[1], and Arthur W. Toga[1]

[1] Lab of Neuro Imaging, UCLA School of Medicine, Los Angeles, CA, USA
{yshi,ztu,ivo.dinov,thompson,toga}@loni.ucla.edu
[2] School of Medicine, Stanford University, Stanford, CA, USA
[3] Harvard Medical School, Boston, MA, USA

Abstract. In this paper we propose an automated approach for joint sulci detection on cortical surfaces by using graphical models and boosting techniques to incorporate shape priors of major sulci and their Markovian relations. For each sulcus, we represent it as a node in the graphical model and associate it with a sample space of candidate curves, which is generated automatically using the Hamilton-Jacobi skeleton of sulcal regions. To take into account individual as well as joint priors about the shape of major sulci, we learn the potential functions of the graphical model using AdaBoost algorithm to select and fuse information from a large set of features. This discriminative approach is especially powerful in capturing the neighboring relations between sulcal lines, which are otherwise hard to be captured by generative models. Using belief propagation, efficient inferencing is then performed on the graphical model to estimate each sulcus as the maximizer of its final belief. On a data set of 40 cortical surfaces, we demonstrate the advantage of joint detection on four major sulci: central, precentral, postcentral and the sylvian fissure.

1 Introduction

Cortical sulci are important landmarks in human brain mapping because they encode rich information about the convolution patterns of human brains[1] and provide guidance for registration tasks[2], but the variability of the brain morphometry poses serious challenges for their automatic detection, thus manual annotation remains the golden standard in practice. In this paper, we propose a novel approach to incorporate prior knowledge from manual tracing by modeling the relation of major sulci with boosting techniques and detect them *jointly* via the solution of an inference problem on graphical models.

Many algorithms were proposed for sulci detection in previous works. Curvature features were used in [3,4,5] to detect sulci semi-automatically with the need of manually inputing the starting and ending points. Depth features with respect to a shrink wrap surface were also used for sulci detection on surfaces[6,7]. The extraction of sulci from volume images were proposed in [8,9,10], but human interactions are still necessary to pick out specific sulci from the results.

N. Karssemeijer and B. Lelieveldt (Eds.): IPMI 2007, LNCS 4584, pp. 98–109, 2007.

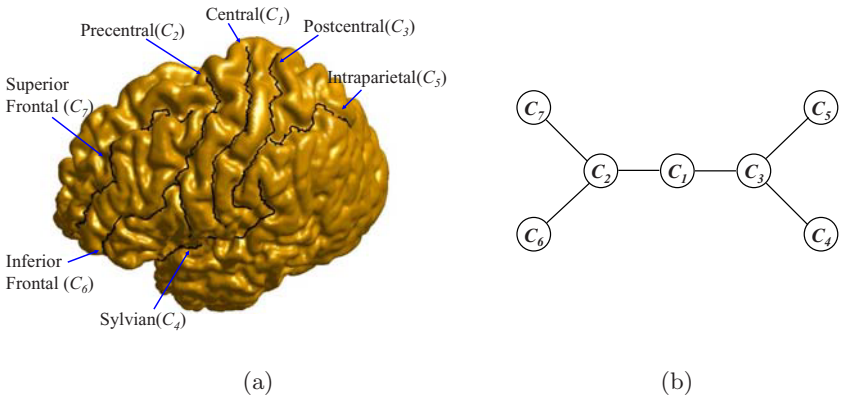


Fig. 1. An illustration of joint sulci detection. (a) Seven major sulci on a cortical surface. (b) A graphical model representing the local dependency between neighboring sulci.

To address this challenge, learning-based approaches were introduced into the detection process. Because of its simplicity, techniques based on the principal component analysis(PCA) of point sets[11] were used in[12,13]. Graphical models and neural networks were used in [14] for automatic recognition of sulcal ribbons, which only identify subsets of major sulci. A learning-based technique based on probabilistic boosting trees[15] was proposed in [16] for the automatic detection of cortical sulci from volume images, but each sulcus was detected separately.

We propose in this work to detect multiple sulci jointly with graphical models. From the experience of manual tracing, this seems a natural choice as the knowledge about the relative location of sulci is frequently utilized for the correct identification of these curves. Modeling the relation of multiple objects has also been shown useful in medical image segmentation[17]. As an example, seven major sulci are plotted on the lateral surface of a cortex in Fig. 1(a). Even though we can see that most parts of sulci follow furrows of high curvature, choices have to be made at intersections of multiple furrows because of the variability of the cortex. What makes this task more difficult is that the gyral regions have to be crossed sometimes to ensure a continuous curve is generated for each sulcus. To counter this kind of complications, protocols are defined in practice on how to use *local* dependency of sulcal lines for manual annotation. For the example in Fig. 1(a), the precentral sulcus has to cross a gyrus to meet the requirement that it should follow a path as parallel as possible to the central sulcus. In determining the inferior portion of postcentral sulcus that is highly variable, its relation with respect to the tail of the sylvian fissure also helps to provide critical information.

There are two main challenges, however, to formulate a tractable inference problem over graphical models for joint sulci detection. (1) With each node of the graph representing a sulcus, the random variables of interest here live in high

dimensional shape spaces and it is generally hard to do inferences directly over such spaces. (2) Every major sulcal curve observes flexible shape and it is hard to capture their individual as well as joint regularity by a good prior.

In our joint detection framework, we tackle these challenges by first constructing a sample space containing a finite number of candidate curves at each node of the graph. These sample spaces greatly reduce the search range of inference algorithms and they are generated automatically based on the Hamilton-Jacobi skeleton of sulcal regions on triangulated cortical surfaces. We then use boosting techniques[18] to learn discriminative shape models of each sulcus and their neighboring relations. The advantages are twofold. (1) The algorithm is able to automatically select and fuse a set of informative shape features, characterizing unary as well as binary relationships of the sulcal curves, from a large set of candidate features. (2) The priors are learned directly from the training data and there is no parameter to tune for different sucal curves. Traditional generative model based algorithms, e.g. PCA, have difficulty in modeling such complicated priors due to its Gaussian assumptions. In making the inference, a belief propagation algorithm[19,20] is used to perform inferences efficiently on these sample spaces for joint sulci detection.

In the rest of the paper, we first propose our joint detection framework with graphical models in section 2. The generation of sample spaces for each node of the graph is then described in section 3. After that, we develop a boosting approach in section 4 to learn potential functions in graphical models for sulci of interest. In section 5, experimental results on the joint detection of four major sulci: central, precentral, postcentral and the sylvian fissure on 40 cortical surfaces are presented. Finally, conclusions are made in section 6.

2 The Joint Detection Framework

For the detection of a set of major sulci $C_1, C_2, \cdots, C_L$ on a cortical surface $\mathcal{M}$, we assume an undirected graphical model $G = (V, E)$ that represents the Markovian relation among sulci, where $V = \{C_1, C_2, \cdots, C_L\}$ are the set of nodes, and E is the set of edges in the graph. As an example, a graphical model for the seven sulci in Fig. 1(a) is shown in Fig. 1(b). Because the number of major sulci is small on the cortical surface, it is straightforward to construct the graph structure of such models and this only needs to be done once for the same detection task.

To perform inferences on sulcal lines with graphical models, it is critical to first specify a proper sample space for each node as the general space of curves is infinite dimensional. One possible solution is to reduce the dimension of shape spaces with PCA, but there is no guarantee that these parameterized curves will live on the cortical surface and follow the bottom of sulcal regions. To overcome this problem, we generate a set of candidate curves S_i automatically for each sulcus C_i using a novel algorithm that will be developed in section 3. These candidate curves are weighted geodesics on the cortical surface and follow the skeleton of sulcal regions closely. By adopting the set of candidate curves S_i as

Table 1. The AdaBoost algorithm[18]

Given training data: $(x_1, y_1, w_1^1), \cdots, (x_n, y_n, w_n^1)$ 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 $t = 1, \cdots, T$

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

the sample space for each node C_i, we are able to model each sulcus as a discrete random variable with values in a finite set and this makes the inference on the graphical model computationally tractable.

Over the sample spaces of all the nodes in a graphical model, we define two types of potential functions to complete the construction of the graphical model: the local evidence function $\phi_i : S_i \to \mathbb{R}$ at each node C_i and the compatibility function $\psi_{i,j} : S_i \times S_j \to \mathbb{R}$ for $(C_i, C_j) \in E$. To incorporate shape priors about individual sulcus and their relations, we propose a discriminative approach using AdaBoost[18] in section 4 to learn both types of potential functions from manually annotated training data. Given input data x, the discriminative method learns the posterior probability $p(y|x)$ of a label y. With the discriminative approach, there is no need of specifying parametric forms for prior models of sulcal lines and their neighboring relations. Instead we use a large set of features derived from training data and selectively combine information from these features with boosting techniques. The central idea of AdaBoost, as listed in Table I, is the formation of a strong classifier through the combination of a series of weak classifiers. Using the decision function f generated from AdaBoost, an approximate posterior can then be defined and used as the potential function.

With the graphical model defined, we can write down the joint distribution of all the sulci as:

$$p(C_1, \cdots, C_L) = \frac{1}{Z} \prod_{(C_i, C_j) \in E} \psi_{i,j}(C_i, C_j) \prod_{C_i \in V} \phi_i(C_i) \tag{1}$$

where Z is the partition function for normalization. For sulci detection, we use belief propagation to compute the marginal distribution of each sulcus from the joint distribution because it is applicable to graphs both with and without cycles. 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

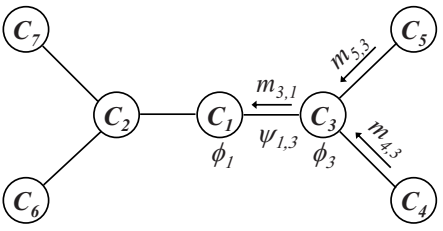


Fig. 2. An example of message passing in the graphical model of Fig. 1(b)

neighbor C_j is defined as:

$$m_{i,j}(C_j) = \sum_{C_i \in S_i} \psi_{i,j}(C_i, C_j)\phi_i(C_i) \prod_{C_k \in \mathcal{N}(C_i) \setminus C_j} m_{k,i}(C_i) \tag{2}$$

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_4 and C_5 to C_3, and then to C_1 in the graphical model shown in Fig. 1(b). If we continue this message passing process until it converges, i.e., when the messages stop changing, we obtain the final belief at each node of the graph as:

$$b_i(C_i) = \phi_i(C_i) \prod_{C_j \in \mathcal{N}(C_i)} m_{j,i}(C_i) \quad i = 1, 2, \cdots, L. \tag{3}$$

Using belief propagation, we collect information from all the nodes to form the final belief for every sulcus. For graphs without cycles[19], such as trees, the final belief function corresponds to the marginal distribution of each sulcus derived from the joint distribution in (1), thus we can detect a sulcus by maximizing the final belief:

$$C_i^* = \arg \max_{C_i \in S_i} b_i(C_i) \quad i = 1, 2, \cdots, L. \tag{4}$$

Even for graphs with cycles, belief propagation is known to frequently perform well and generate excellent results[20], most notably for its near Shannon limit performance in turbo decoding[21]. Thus the above equation is also applicable for joint sulci detection in the case that the graphical model has loops.

3 Generation of Sample Spaces

Given a cortical surface, there are three main stages in our algorithm of sample space generation for each node of a graphical model: skeletonization of sulcal regions, a learning-based approach that picks out candidates for the starting and ending point of the sulcus, and the generation of member curves of the sample space as weighted geodesics on the cortical surface.

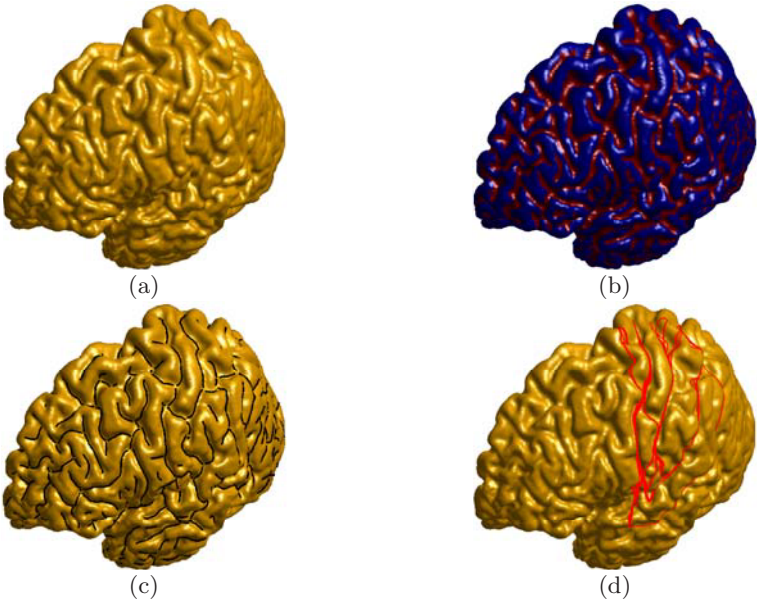


Fig. 3. The process of generating sample spaces for sulci detection. (a) The original cortical surface. (b) Mesh segmentation results. Red: sulcal regions; blue: gyral regions. (c) The skeleton of sulcal regions plotted in black. (d) Curves from the sample space of the central sulcus plotted in red.

In the first stage, we compute the skeleton of sulcal regions with the method we reported in[22]. For completeness we briefly review the main steps of this algorithm. We represent the cortical surface $\mathcal{M}$ as a triangular mesh and first compute the principle curvatures at each vertex. Using the curvature features, the cortical surface is then partitioned into sulcal and gyral regions using graph cuts. As an illustration, we show in Fig. 3(b) the result of the partition algorithm for the cortical surface in Fig. 3(a). Finally an extension of the method of Hamilton-Jacobi skeleton[23] is used to calculate the skeleton of sulcal regions as shown in Fig. 3(c). For all vertices of $\mathcal{M}$ in the skeleton, we classify them into three types using the neighboring system of the triangular mesh:

- End points: vertices with one neighbor in the skeleton.
- Knot points: vertices with three or more neighbors in the skeleton.
- Middle points: vertices with two neighbors in the skeleton.

We learn a two-class classifier with AdaBoost in the second stage to pick a set of candidate points from the union of end and knot points in the skeleton for the starting and ending point of each sulcus. We use a set of K cortical surfaces with manually labeled sulci for the construction of our training data. These cortical surfaces are assumed to be registered to a common coordinate space such as ICBM. To learn the classifier for the starting point of a sulcus C_i, we construct

the training data as follows. The starting point of the manually labeled sulcus C_i on the K cortical surfaces are assigned label 1 and weight $1/2K$. The union of all the knot and end points from the skeletons of the K cortical surfaces are assigned label -1 and weight $1/2\tilde{N}$ where $\tilde{N}$ is the total number of points in this set. At each iteration of AdaBoost, we train a perceptron as the weak classifier using the pocket algorithm with ratchet[24]. As shown in Table I, the final decision function generated from AdaBoost is of the following form:

$$f = \sum_{t=1}^{T} \alpha_t h_t \tag{5}$$

where $h_t : \mathbb{R}^3 \to \{-1, 1\}$ is the weak classifier at the t-th iteration of AdaBoost and α_t is the weight for this weak classifier. Applying the decision function f to all the end and knot points in the skeleton of a cortical surface, we pick the candidates for the starting point of C_i as M points with the largest f values. Typically we choose $M = 10$ in our experiments and this generates a sufficiently large candidate set for the final detection of sulcal lines according to our experience. Similarly, a candidate set of M points can also be generated for the ending point of C_i.

Given the candidate set for the starting and ending points of the sulcal line C_i, we connect them with weighted geodesics on $\mathcal{M}$ to generate the sample space S_i. The weights are derived from the geodesic distance transform d of the skeleton on $\mathcal{M}$ generated in the first stage. We compute these geodesics with the fast marching algorithm on triangular meshes[25]. For a candidate of starting point X_s and a candidate for the ending point X_e, we connect them through a weighted geodesic with the weight defined as

$$F = e^{-d^2/2\sigma^2} \tag{6}$$

to encourage the weighted geodesic overlapping as much as possible with the skeleton of sulcal regions. To find this geodesic, we first compute a weighted distance transform d^w on $\mathcal{M}$ by solving the Eikonal equation

$$\nabla d^w F = 1 \tag{7}$$

intrinsically over the cortical surface and trace backward from X_e to X_s in gradient descent directions. This geodesic is then added to the sample space S_i. We choose the parameter $\sigma = 1, 2, 3, 4, 5$ to cover a wide range of possible routes. So overall the sample space S_i of C_i is composed of $5M^2$ curves. As an illustration, all the candidate curves in the sample space of the central sulcus on the cortical surface in Fig. 1(a) are plotted in Fig. 1(d) and we can see that the true central sulcus is included in this sample space.

4 Learning Potential Functions Using AdaBoost

In this section, we describe our learning-based approach for the construction of the potential functions in graphical models to take into account individual

and joint shape priors of major sulci. Our approach is discriminative and learns both the local evidence functions ϕ_i and the compatibility functions $\psi_{i,j}$ with AdaBoost. For each potential function, we construct a set of features and learn a strong two-class classifier by combining with AdaBoost a series of weak classifiers on these features. It is shown in [26] that AdaBoost approximates logistic regression and its result can be used to estimate the probability of a class label, which we then use to define the potential function.

For a sulcus C_i we learn its local evidence function ϕ_i from a training data set of K cortical surfaces with the sulcus manually labeled on each surface. We also assume these cortical surfaces are registered to the same common coordinate system as in section 3. For the same sulcus, we generate a sample space on each of the cortical surface in the training data. All curves are parameterized with N (typically 100) uniformly sampled points such that one-to-one correspondences are established between curves. In the training data set, all manually labeled sulcus for C_i on the K cortical surfaces are assigned label 1 and weight $1/2K$ and all the curves in the sample spaces are assigned a label -1 and weight $1/10KM^2$. The same perceptron in section 3 is used as our weak classifier. At each iteration of AdaBoost, we train N perceptrons $P_n(1 \leq n \leq N)$ with the n-th point from all the curves and pick the one with the best performance as the weak classifier h_t. By combining all the weak classifiers, we obtain the final decision function $f(C_i) = \sum_{t=1}^{T} \alpha_t h_t(C_i)$. Following [26], we define the local evidence function for C_i as:

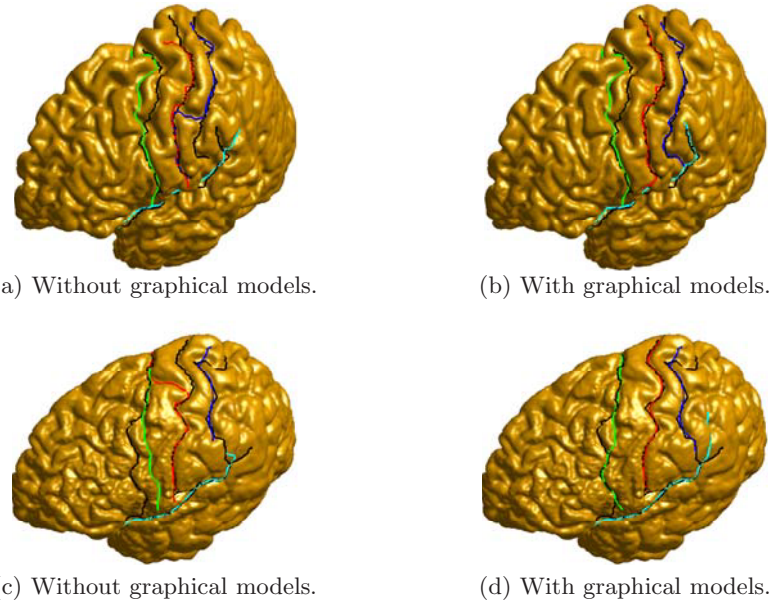
$$\phi_i(C_i) = \frac{e^{2f(C_i)}}{1 + e^{2f(C_i)}} \quad \forall C_i \in S_i. \tag{8}$$

The local evidence function approaches the value 1 at curves in the sample space that are similar to the manually labeled sulcus in the training data, indicating intuitively that they exhibit strong evidence to be the sulcus we want to detect.

To learn the compatibility function between two sulci C_i and C_j, we define the feature used for training as $d_{i,j} = C_i - C_j$, which is the difference between the two curves. For each cortical surface in the training data, the manually labeled sulci for C_i and C_j are used to construct the feature vector $d_{i,j}$ with label 1 and weight $1/2K$. From the sample spaces of C_i and C_j on the K cortical surfaces, we randomly pick $5KM^2$ pairs to construct the negative sample data with label -1 and weight $1/10KM^2$. The same boosting process as above is applied to learn a decision function for the compatibility between the two sulci $f(C_i, C_j) = \sum_{t=1}^{T} \alpha_t h_t(d_{i,j})$ and we can define the compatibility function between C_i and C_j as:

$$\psi_{i,j}(C_i, C_j) = \frac{e^{2f(C_i,C_j)}}{1 + e^{2f(C_i,C_j)}} \quad \forall C_i \in S_i, C_j \in S_j. \tag{9}$$

With AdaBoost, we have developed a common solution for the learning of both local evidence functions and compatibility functions. Even though we used only features derived from coordinate information in our current work, the boosting method for learning priors is general and allows the inclusion of more features in future work. By inserting the potential functions into graphical models, the belief propagation process can then be used for the joint detection of major sulci.



(a) Without graphical models. (b) With graphical models.

(c) Without graphical models. (d) With graphical models.

Fig. 4. Sulci detection results on two cortical surfaces. (a) and (b) show results from a surface in the training data. (c) and (d) show results from one of the surface in the testing data. For all the results, manually labeled results are plotted in black for comparison. The detected curves for the central, precentral, postcentral, and sylvian fissure are plotted in red, green, blue and cyan.

5 Experimental Results

In this section we present preliminary experimental results for the joint detection of four major sulci: the central, precentral, postcentral sulcus and sylvian fissure. The graphical model used for the joint detection of these four sulci is the subgraph in Fig. 1(b) that includes the nodes C_1, C_2, C_3 and C_4.

In our experiments, we have a dataset of 40 cortical surfaces with the four sulci manually labeled. We used 20 of them as the training data and the other 20 for testing. During the training stage, we first computed the skeleton of sulcal regions for the 20 training data. Decision functions were then learned for each sulcus such that a set of candidate points can be generated for both of its starting and ending point. After that, a sample space was generated for each sulcus with weighted geodesics as described in section 3. With these sample spaces, the local evidence functions and the compatibility functions between neighboring vertices in the graphical model were learned using the boosting method developed in section 4. For every cortical surface in the testing data, the sample space of each sulcus was generated with the same classifier learned from the training data.

Once all the potential functions and sample spaces were constructed, we applied the joint detection algorithm with graphical models in section 2 to detect

the four sulci on all 40 cortical surfaces. As a comparison, we also detected each sulcus independently without using graphical models. This is realized by simply picking the curve in the sample space that maximizes the local evidence function. To illustrate the advantage of the joint detection algorithm, we show in Fig. 4 the detection results on two cortical surfaces, one from the training data and the other from the testing data, using these two different methods. For the first surface, the results without using graphical models is shown in Fig. 4(a) and we can see the inferior part of the postcentral sulcus overlaps with the central sulcus and this is clearly undesirable. In the joint detection results shown in Fig. 4(b), it is easy to see the postcentral sulcus is correctly detected. Thanks to the compatibility functions learned with boosting, the tail of the sylvian fissure detected in Fig. 4(b) also agrees better with manually labeled results than the sylvian fissure detected in Fig. 4(a). In results of the second surface shown in Fig. 4(c) and (d), improved results were also obtained with graphical models for the central and precentral sulcus as compared to the results detected independently.

We next perform a quantitative analysis of the performance of our sulci detection algorithm by comparing with manually labeled curves. For two curves $\tilde{C}_1$ and $\tilde{C}_2$ that are parameterized with N points, we define two distances:

$$D_a(\tilde{C}_1, \tilde{C}_2) = \frac{1}{N} \sum_{x_n \in \tilde{C}_1} \min_{y_m \in \tilde{C}_2} \parallel x_n - y_m \parallel \tag{10}$$

and

$$D_h(\tilde{C}_1, \tilde{C}_2) = \max_{x_n \in \tilde{C}_1} \min_{y_m \in \tilde{C}_2} \parallel x_n - y_m \parallel . \tag{11}$$

The distance D_a is an average of the distance from points on $\tilde{C}_1$ to $\tilde{C}_2$, while D_h measures the Hausdorff distance from $\tilde{C}_1$ to $\tilde{C}_2$. For a detected sulcus C_i and its manually labeled result $\overline{C}_i$, we use four distances to measure the difference between them: $D_a^1 = D_a(C_i, \overline{C}_i)$, $D_a^2 = D_a(\overline{C}_i, C_i)$, $D_h^1 = D_h(C_i, \overline{C}_i)$, and $D_h^2 = D_h(\overline{C}_i, C_i)$.

Quantitative comparisons to manually labeled results were performed for both the sulci detected jointly with graphical models and independently without graphical models. For these two different groups of results, the average of the four distances over the training and testing data were computed and listed separately for each sulcus in Table 2. Overall we have 32 average distances for each group of results. The advantage of the jointly detected results are clear as they outperform results without graphical models in 28 of the 32 distances. A more detailed analysis shows that we achieve very good performance with our joint sulci detection algorithm in terms of D_a^1 and D_a^2 for all four sulci, and the results are especially encouraging for the central sulcus where an average distance of around 1mm is obtained to the manually annotated curves. The relatively large distances in terms of D_h^1 and D_h^2 are mostly due to the variability of the starting and ending part of sulci. Considering the lack of consensus among manual tracers on these parts, it might be interesting to use other metrics in our future research to measure how well the main body of each sulcal curve is detected.

Table 2. Differences between sulci detected jointly with graphical models and independently as compared to manually annotated results (in millimeters)

	Without graphical models				With graphical models			
Sulci	D_a^1	D_a^2	D_h^1	D_h^2	D_a^1	D_a^2	D_h^1	D_h^2
Central (training)	1.86	1.91	7.88	8.40	0.99	1.36	5.64	6.98
Precentral (training)	3.01	2.50	11.48	11.87	2.57	2.56	10.03	11.13
Postcentral(training)	5.65	5.00	15.34	14.58	2.90	3.26	11.75	12.19
Sylvian (training)	2.68	2.55	11.14	13.05	2.88	2.32	11.15	10.83
Central (testing)	3.04	2.98	10.52	11.01	1.10	1.59	6.59	8.55
Precentral (testing)	4.37	4.48	13.19	14.75	2.92	3.41	10.55	12.60
Postcentral (testing)	2.80	3.10	11.33	11.80	2.50	3.04	10.16	12.42
Sylvian (testing)	3.07	2.75	13.41	13.59	2.65	2.48	12.09	12.55

6 Conclusions

A general framework for the joint detection of major sulci was proposed in this paper. Using boosting techniques, we integrated discriminative shape priors of each sulcus and their Markovian relations into graphical models and transformed sulci detection into a tractable inference problem over sample spaces of candidate curves. The boosting approach is flexible and allows the inclusion of new features to capture more detailed priors in future work. We are also testing the joint detection of more major sulci with our algorithm.

Acknowledgment

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).

References

1. Ono, M., Kubik, S., Abarnathey, C.: Atlas of the Cerebral Sulci. Thieme Medical Publishers (1990)
2. Thompson, P.M., Hayashi, K.M., Sowell, E.R., Gogtay, N., Giedd, J.N., Rapoport, J.L., de Zubicaray, G.I., Janke, A.L., Rose, S.E., Semple, J., Doddrell, D.M., Wang, Y., van Erp, T.G.M., Cannon, T.D., Toga, A.W.: Mapping cortical change in alzheimers disease, brain development, and schizophrenia. NeuroImage 23, S2–S18 (2004)
3. Khaneja, N., Miller, M., Grenander, U.: Dynamic programming generation of curves on brain surfaces. IEEE Trans. Pattern Anal. Machine Intell. 20(11), 1260–1265 (1998)
4. Bartesaghi, A., Sapiro, G.: A system for the generation of curves on 3D brain images. Human Brain Mapping 14, 1–15 (2001)
5. Lui, L.M., Wang, Y., Chan, T.F., Thompson, P.M.: Automatic landmark and its application to the optimization of brain conformal mapping. In: Proc. CVPR. vol. 2, pp. 1784–1792 (2006)
6. Rettmann, M.E., Han, X., Xu, C., Prince, J.L.: Automated sulcal segmentation using watersheds on the cortical surface. NeuroImage 15(2), 329–244 (2002)

7. Kao, C., Hofer, M., Sapiro, G., Stern, J., Rotternberg, D.: A geometric method for automatic extraction of sulcal fundi. In: Proc. ISBI 2006, pp. 1168–1171 (2006)
8. Lohmann, G.: Extracting line representations of sulcal and gyral patterns in MR images of the human brain. IEEE Trans. Med. Imag. 17(6), 1040–1048 (1998)
9. Zhou, Y., Thompson, P.M., Toga, A.W.: Extracting and representing the cortical sulci. IEEE Computer Graphics and Applications 19(3), 49–55 (1999)
10. Mangin, J.F., Frouin, V., Bloch, I., Régis, J., López-Krahe, J.: From 3d magnetic resonance images to structural representations of the cortex topography using topology preserving deformations. Journal of Mathematical Imaging and Vision 5(4), 297–318 (1995)
11. Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models-their training and application. Computer Vision and Image Understanding 61(1), 38–59 (1995)
12. Lohmann, G., Cramon, D.: Automatic labelling of the human cortical surface using sulcal basins. Medical Image Analysis 4, 179–188 (2000)
13. Tao, X., Prince, J., Davatzikos, C.: Using a statistical shape model to extract sulcal curves on the outer cortex of the human brain. IEEE Trans. Med. Imag. 21(5), 513–524 (2002)
14. Rivière, D., Mangin, J., Papadopoulos-Orfanos, D., Martinez, J., Frouin, V., Régis, J.: Automatic recognition of cortical sulci of the human brain using a congregation of neural networks. Medical Image Analysis 6, 77–92 (2002)
15. Tu, Z.: Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering. In: Proc. ICCV 2005, vol. 2, pp. 1589–1596 (2005)
16. Zheng, S., Tu, Z., Yuille, A., Reiss, A., Dutton, R., Lee, A., Galaburda, A., Thompson, P., Dinov, I., Toga, A.: A learning-based algorithm for automatic extraction of the cortical sulci. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) MICCAI 2006. LNCS, vol. 4190, pp. 695–703. Springer, Heidelberg (2006)
17. Pizer, S., Jeong, J., Lu, C., Joshi, S.: Estimating the statistics of multi-object anatomic geometry using inter-object relationships. In: Proc. DSSCV 2005, pp. 60–71 (2005)
18. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
19. Perl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufman, San Mateo (1988)
20. Yedidia, J., Freeman, W., Weiss, Y.: Understanding Belief Propagation and Its Generalizations, pp. 239–269. Morgan Kaufmann Publishers Inc, San Francisco (2003)
21. Berrou, C., Glavieus, A., Thitimajshima, P.: Near Shannon limit error-correcting coding and decoding: Turbo-codes. In: Proc. IEEE Int. Conf. on Communications, pp. 1064–1070 (1993)
22. Shi, Y., Reiss, A., Lee, A., Dutton, R., Bellugi, U., Galaburda, A., Korenberg, J., Mills, D., Dinov, I., Thompson, P., Toga, A.: Hamilton-Jacobi skeletons on cortical surfaces with applications in characterizing the gyrification pattern in Williams syndrome. In: Proc. ISBI 2007 (2007)
23. Siddiqi, K., Bouix, S., Tannebaum, A., Zuker, S.: Hamilton-Jacobi skeletons. Int'l Journal of Computer Vision 48(3), 215–231 (2002)
24. Gallant, S.: Perceptron-based learning algorithms. IEEE Trans. Neural Networks 1(2), 179–191 (1990)
25. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. Proc. Natl. Acad. Sci. USA 95(15), 8431–8435 (1998)
26. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Ann. Statist. 28(2), 337–407 (2000)