Title
Automated Extraction of the Cortical Sulci Based on a Supervised Learning Approach

Permalink
https://escholarship.org/uc/item/1jh5m04n

Authors
Tu, Zhuowen
Zheng, Songfeng
Yuille, Alan L.
et al.

Publication Date
2007-07-09

Peer reviewed

Automated Extraction of the Cortical Sulci Based on a Supervised Learning Approach

Zhuowen Tu, Songfeng Zheng, Alan L. Yuille, Allan L. Reiss, Rebecca A. Dutton, Agatha D. Lee, Albert M. Galaburda, Ivo Dinov, Paul M. Thompson, and Arthur W. Toga*

Abstract—**It is important to detect and extract the major cortical sulci from brain images, but manually annotating these sulci is a time-consuming task and requires the labeler to follow complex protocols [1], [2]. This paper proposes a learning-based algorithm for automated extraction of the major cortical sulci from magnetic resonance imaging (MRI) volumes and cortical surfaces. Unlike alternative methods for detecting the major cortical sulci, which use a small number of predefined rules based on properties of the cortical surface such as the mean curvature, our approach learns a discriminative model using the probabilistic boosting tree algorithm (PBT) [3]. PBT is a supervised learning approach which selects and combines hundreds of features at different scales, such as curvatures, gradients and shape index. Our method can be applied to either MRI volumes or cortical surfaces. It first outputs a probability map which indicates how likely each voxel lies on a major sulcal curve. Next, it applies dynamic programming to extract the best curve based on the probability map and a shape prior. The algorithm has almost no parameters to tune for extracting different major sulci. It is very fast (it runs in under 1 min per sulcus including the time to compute the discriminative models) due to efficient implementation of the features (e.g., using the integral volume to rapidly compute the responses of 3-D Haar filters). Because the algorithm can be applied to MRI volumes directly, there is no need to perform preprocessing such as tissue segmentation or mapping to a canonical space. The learning aspect of our approach makes the system very flexible and general. For illustration, we use volumes of the right hemisphere with several major cortical sulci manually labeled. The algorithm is tested on two groups of data, including some brains from patients with Williams Syndrome, and the results are very encouraging.**

Index Terms—**Cortical sulci, discriminative models, dynamic programming, learning, magnetic resonance (MR) images, probability boosting tree.**

Z. Tu, R. A. Dutton, A. D. Lee, I. Dinov, and P. M. Thompson are with the Laboratory of Neuro Imaging, University of California-Los Angeles, School of Medicine, Los Angeles, CA 90095 USA.

*A. W. Toga is with the Laboratory of Neuro Imaging, University of California-Los Angeles, School of Medicine, Los Angeles, CA 90095 USA (e-mail: toga@loni.ucla.edu).

S. Zheng and A. L. Yuille are with Department of Statistics, University of California-Los Angeles, 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 School of Medicine, Harvard University, Cambridge, MA 02138 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

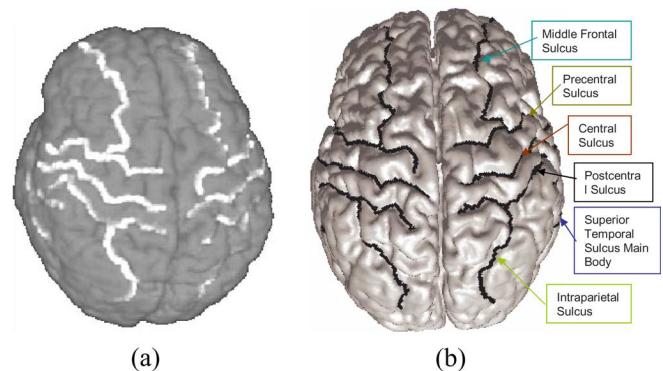Digital Object Identifier 10.1109/TMI.2007.892506

Fig. 1. Examples of cortical sulci: (a) shows an MRI volume overlayed with several major cortical sulci, such as the Central Sulcus and the Postcentral Sulcus; (b) illustrates a corresponding extracted surface with the same set of manually labeled sulcal curves.

I. INTRODUCTION

CORTICAL sulci are important structures of the brain serving as landmarks associated with major functional regions. Reliably extracting major cortical sulci from magnetic resonance (MR) images helps us to better understand the functions of the brain [4], assists registration tasks in brain mapping [5], [6], and facilitates studies for discovering brain diseases and monitoring brain growth [7]. Fig. 1 shows several major cortical sulci: the Central Sulcus, the Postcentral Sulcus, the Superior Temporal Sulcus, the Intraparietal Sulcus, the Middle Frontal Sulcus, and the Intraparietal Sulcus.

Cortical sulci lie on the valleys of cortical folds, and can be characterized by mathematical measures such as mean curvatures [8]–[11]. But detecting cortical sulci is difficult because they have complicated geometric and photometric patterns in MR images which are hard to distinguish from similar patterns caused by other brain structures (e.g., other folds in the brain). Even expert anatomists have to use complicated protocols, involving high-level knowledge, to precisely locate and delineate them [1], [2], [12]. In general, manually delineating the major cortical sulci is a difficult and time-consuming task.

Most existing algorithms for automatic detection of cortical sulci [4], [8], [13]–[15] use cortical surfaces as input, which require a preprocessing stage to segment the tissue.[1] These algorithms use geometric information in different ways.

---

[1]In this paper, we are interested in extracting the sulcal centerlines. We refer to them as sulci here.

Tao *et al.* [13] used global shape priors, obtained by principal component analysis (PCA), combined with local measures such as mean curvature. Their method involved mapping the cortical surface to the unit sphere. Khaneja *et al.* [8] defined a shape model of the sulcal curves in terms of local geometric properties, e.g., curvature and torsion, and used a dynamic programming algorithm to find the curves by minimizing an energy function. This algorithm was not fully automatic since it required the starting and ending points of the sulci to be specified by hand. Vaillant and Davatzikos [15] used an active contour model for extracting the sulcal curve which also needed careful user initialization. In [14], the major sulci are detected based on graph matching using multilayer neural networks, which are trained to learn the cortical folding patterns. The work by [16] extracted sulcal regions using a watershed transformation method applied to cortical surfaces.

In this work, we propose an algorithm for automated extraction of the major cortical sulci from magnetic resonance imaging (MRI) volumes or their cortical surfaces. The key to our approach is to estimate the likelihood of each voxel being on a major sulcal curve, or at the starting and ending points of these curves (see next paragraph for how these likelihoods are learned). Our detection algorithm consists of three steps:

1) Given input data, either a volume or an extracted surface, it computes the likelihood of each voxel being on a major sulcal curve by looking at a subvolume in the data ($15 \times 15 \times 15$) centered at this voxel. (In the paper, we use the extracted surfaces and volumes reported in [6] which are T1-weighted brain MRI scans and we use resampled isotropic volumes with resolution $1\,\mathrm{mm} \times 1\,\mathrm{mm} \times 1\,\mathrm{mm}$).

2) The starting and ending points are automatically detected from a similar likelihood.

3) The algorithm outputs the major sulcal curves using the dynamic programming algorithm to minimize an energy function which combines the likelihood computed in (1) with a shape prior.

We learn the likelihoods by the *probabilistic boosting tree* (PBT) method [3] which requires manually labeled data as input. PBT selects a set of features from a large pool of candidates and combines them into a hierarchical structure. The candidate pool consists of around 8000 features at three scales including gradients, curvatures, shape index, locations, and 3-D Haar filters. PBT has few parameters, is highly flexible and adaptive, and works on extracted surfaces as well as on MRI volumes (for which, obviously, cortical surface segmentation is not required). Avoiding intermediate stages, such as surface segmentation or mapping to a canonical sphere, is desirable since it reduces the chance of introducing errors and simplifies the algorithm significantly. We also use PBT to detect the starting and ending points of the sulci.

The algorithm is fully automatic, very general, and has almost no parameter to tune for different major sulcal curves. Moreover, it can be applied in other curve/object detection tasks in medical imaging. See [17] for our preliminary results.

We note that there are existing edge detection methods in the image processing community which are also learning based [18]–[20]. Papers [18] and [19] proposed methods to detect edges in 2-D natural images, but these approaches merely perform cue combination to fuse several carefully designed image features, and it is not clear how to extend those methods to perform cortical sulci detection. The learning strategy in our algorithm is more similar in spirit to [20], but the method in [20] only computes a likelihood map for edges and contours in 2-D natural images (i.e., there is no explicit curve detection stage). Geman and Jedynak [21] proposed an algorithm for detecting roads using a shape prior and a learning method similar to [18].

For learning and evaluation, we first use a dataset of 40 images of the right hemisphere with several major cortical sulci manually labeled (we refer to images as either volumes or extracted surfaces in the paper; they are not slices of a 3-D volume). We split them randomly into 15 training images and 25 testing images. We also used an additional dataset of 40 images from patients with Williams Syndrome (WS), where the sulci are also manually delineated by image analysts trained in cortical neuroanatomy (details of the process of extracting the major sulci are reported in [6]) We tested our algorithm on the 15 training images and 25 testing images from the normal dataset, and then on the 40 brain images with WS. For all these cases, we trained and tested our algorithm on both MRI volumes and cortical surfaces (the cortical surfaces were extracted by the algorithm in [22]). We evaluated our algorithm by computing performance measures which compares the output with ground truth (i.e., the manual labels). Our results are very encouraging.

The rest of this paper is organized as follows. An energy minimization framework for sulci detection is developed in Section II. Section III describes the training and testing process of PBT and Section IV gives the overall algorithm. The details of the experiments on the normal dataset are given in Section V, along with the results and performance evaluation. In Section V, we also show and discuss the results for the brains with WS. Finally, we conclude this paper with some remarks about future work in Section VI.

II. PROBLEM FORMULATION

In this section, we formulate the task of detecting 3-D sulcal curves from a 3-D MRI volume $\mathbf{V}$. If the cortical surface is known, we replace the volume by the signal-distance function of the surface map.

The task of cortical sulcal curve detection is to extract a 3-D curve C from an input volume $\mathbf{V}$ defined on a 3-D domain Λ. $\mathbf{V}$ consists of a set of voxels with positions $\mathbf{r} = (x, y, z) \in \Lambda$. We define a neighborhood structure $N(\mathbf{r})$, which is symmetric so that $\mathbf{s} \in N(\mathbf{r})$ if and only if $\mathbf{r} \in N(\mathbf{s})$.

We represent the curve C by a chain of voxels

$$C = \{\mathbf{r}_i, i = 0, \cdots, L\}$$

where L is the length of the curve, and $\mathbf{r}_i$ is the coordinates of the ith voxel on the curve C. These voxels must be adjacent to each other.

We define the background voxels B so that $\Lambda = C \cup B$ and $B \cap C = \emptyset$. We also define an indicator variable $y(\mathbf{r})$ so that $y(\mathbf{r}) = -1$ if $\mathbf{r} \in B$ and $y(\mathbf{r}) = +1$ if $\mathbf{r} \in C$.

We now formulate the task of sulcal curve detection as a probabilistic inference problem: to estimate

$C^* = \arg\max p(C|\mathbf{V})$ where $p(C|\mathbf{V})$ is the posterior probability distribution. Equivalently, we minimize the energy function $E(C;\mathbf{V}) = -\log p(C|\mathbf{V})$.

The probability distribution, and the energy function, are defined by analogy to a Bayesian formulation with some approximations. In a full Bayesian formulation, we would express $p(C|\mathbf{V}) \propto p(\mathbf{V}|C)p(C)$ where $p(\mathbf{V}|C)$ and $p(C)$ are the likelihood and the prior, respectively. But it is currently impractical to specify these distributions accurately, hence our likelihood will involve approximations and our prior will have data dependence.

We define the energy function by

$$E(C;\mathbf{V}) = E_{\text{dis}}(C;\mathbf{V}) + E_{\text{shape}}(C;\mathbf{V}), \quad (1)$$

where $E_{\text{dis}}(C;\mathbf{V})$ corresponds to the likelihood model $-\log p(\mathbf{V}|C)$ and $E_{\text{shape}}(C;\mathbf{V})$ is analogous to the prior $-\log p(C)$.

Firstly, we define $E_{\text{dis}}(C;\mathbf{V})$ by generalizing the pseudo-likelihood function [23]:

$$E_{\text{dis}}(C;\mathbf{V}) = -\sum_{\mathbf{r}\in B}\log p\left(\mathbf{V}(\mathbf{r}),y(\mathbf{r})=-1|\mathbf{V}\left(N(\mathbf{r})/\mathbf{r}\right)\right)$$
$$-\sum_{\mathbf{r}\in C}\log p\left(\mathbf{V}(\mathbf{r}),y(\mathbf{r})=+1|\mathbf{V}\left(N(\mathbf{r})/\mathbf{r}\right)\right) \quad (2)$$

where $\mathbf{V}(\cdot)$ is the intensity value(s) at the given voxel(s); $N(\mathbf{r})$ is the neighborhood centered on voxel $\mathbf{r}$, $N(\mathbf{r})/\mathbf{r}$ includes all the voxels in the neighborhood except $\mathbf{r}$; $p(\mathbf{V}(\mathbf{r}),y(\mathbf{r})|\mathbf{V}(N(\mathbf{r})/\mathbf{r}))$ is a conditional joint probability. Note that (2) is standard pseudo-likelihood [23] if the variable $y(\mathbf{r})$ is removed.

We can reexpress the likelihood function as follows:

$$E_{\text{dis}}(C;\mathbf{V}) = -\sum_{\mathbf{r}\in\Lambda}\log p\left(\mathbf{V}(\mathbf{r}),y(\mathbf{r})=-1|\mathbf{V}\left(N(\mathbf{r})/\mathbf{r}\right)\right)$$
$$-\sum_{\mathbf{r}\in C}\log\frac{p(y(\mathbf{r})=+1|\mathbf{V}(N(\mathbf{r})))}{p(y(\mathbf{r})=-1|\mathbf{V}(N(\mathbf{r})))}. \quad (3)$$

This is done by adding

$$-\sum_{\mathbf{r}\in C}\log p\left(\mathbf{V}(\mathbf{r}),y(\mathbf{r})=-1|\mathbf{V}\left(N(\mathbf{r})/\mathbf{r}\right)\right)$$

into the first term in the right side of (2) and subtracting it from the second term in the right side of (2). The first term in (3) does not depend on C and hence can be ignored. We write

$$E_{\text{dis}}(C;\mathbf{V}) = -\sum_{\mathbf{r}\in C}\log\frac{p(y(\mathbf{r})=+1|\mathbf{V}(N(\mathbf{r})))}{p(y(\mathbf{r})=-1|\mathbf{V}(N(\mathbf{r})))}. \quad (4)$$

where $p(y(\mathbf{r})=+1|\mathbf{V}(N(\mathbf{r})))$ is the posterior probability of a voxel $\mathbf{r}$ belonging to the foreground (sulcal curve) given the neighborhood centered at $\mathbf{r}$. The probabilities $p(y(\mathbf{r})=+1|\mathbf{V}(N(\mathbf{r})))$ and $p(y(\mathbf{r})=-1|\mathbf{V}(N(\mathbf{r})))$ are learned by PBT (see Section III). The second column in Fig. 7, 8, 9 and 10 show some examples of $p(y(\mathbf{r})=+1|\mathbf{V}(N(\mathbf{r})))$ on real MRI volumes.
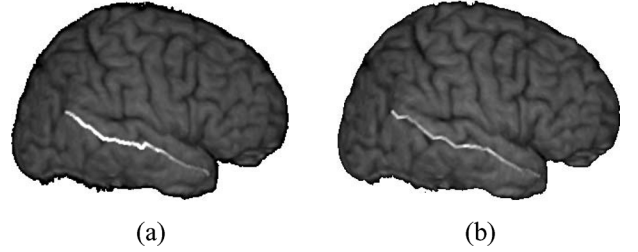

Fig. 2. Illustration of the use of prior terms: (a) shows a sulcal curve detected using the shape prior E_{shape}; (b) illustrates the detection result without the shape prior. The result in (a) is more faithful to the shape of the sulcal curve than that in (b).

Second, we define $E_{\text{shape}}(C;\mathbf{V})$ to be

$$E_{\text{shape}}(C;\mathbf{V}) = -\alpha L + \beta\sum_{i=0}^{L-1}\|\nabla\mathbf{V}(\mathbf{r}_{i+1}) - \nabla\mathbf{V}(\mathbf{r}_i)\| \quad (5)$$

where α and β are positive parameters, and ∇ is the gradient operator. $E_{\text{shape}}(C;\mathbf{V})$ is similar to a standard shape prior, as used in active contour models [24], but differs by being dependent on the data $\mathbf{V}$ (we tried standard data-independent priors, but they were far less effective). The first term in $E_{\text{shape}}(C;\mathbf{V})$ encourages long curves and will also bias toward non-smooth curves; the second term involves data dependent cues, i.e., we prefer the gradients along the detected curve not to change too much.

α and β can be learned based on the training set. Since some local shape information has been implicitly carried out in the discriminative model in (4), they play rather a minor role in the energy function. The effect of $E_{\text{shape}}(C;\mathbf{V})$ can be observed by comparing detection results with and without this shape prior (see Fig. 2). Observe that the curve detected using the shape prior is less smooth and more faithful to the shape of the sulcal curve. Increasing α results in more jagged curves and increasing β encourages the curves stay on smoothly changing gradients.

In summary, the problem of major sulcal curve detection is formulated in terms of finding the curve C^* which minimizes the energy $E(C;\mathbf{V})$ defined in (1).

We use dynamic programming (DP) [25] to minimize the energy $E(C;\mathbf{V})$. DP is guaranteed to find the global minimum, but it requires us to specify the starting and ending points. We will discuss how to specify them automatically in Section IV.

III. LEARNING AND COMPUTING THE DISCRIMINATIVE MODELS

As we can see from (1), (4), and (5), the key in this paper is to learn and compute the discriminative models $p(y(\mathbf{r})|\mathbf{V}(N(\mathbf{r})))$ where $\mathbf{V}(N(\mathbf{r}))$ is the subvolume centered at $\mathbf{r}$.

A. Learning PBT

Given a set of training volumes, along with a major sulcal curve manually labeled in each, we sample positive and negative subvolumes centered on sulcal curves and the background, respectively. The number of samples from a single volume is equal to the number of locations in the volume (although typically the majority of locations contain negative samples). We often have a large training set (e.g., in the order of 10^8.

Learning an accurate decision boundary between the voxels on the curve and the voxels on the background is difficult, and we would like to use as much data as possible to achieve good generalization to new data.

Advances in machine learning, such as the boosting algorithm [26], have allowed us to take advantage of the size of this high-dimensional dataset. We choose to use the probabilistic boosting tree (PBT) algorithm proposed in [3] to efficiently capture the large variation of the foreground and the background patterns in a divide-and-conquer strategy. The advantages of the PBT over the traditional boosting algorithms can be found in [3]. PBT can be seen as a combination of a decision tree with boosting (a cascade is, then, just a special case of a tree). In this section, we suppose that $\mathbf{x}$ is the input vector and y is the corresponding label (in the experiments, $\mathbf{x}$ will be $\mathbf{V}(N(\mathbf{r}))$). For notational simplicity, we denote the probability computed by each boosting node as

$$q(y|\mathbf{x}) = \frac{\exp\{2yH(\mathbf{x})\}}{1 + \exp\{2yH(\mathbf{x})\}} \quad \text{with} \quad y \in \{+1, -1\} \quad (6)$$

where $H(\mathbf{x})$ is the trained strong classifier at the node, with

$$H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

in which $h_t(\mathbf{x})$ is a weak classifier. The training process of boosting on each node follows the standard way of learning the parameters α_t in [26]. In the context of boosting in machine learning, a weak classifier refers to any classifier which is better than a random guess. A strong classifier is one with good classification results (e.g., with error ≤ 0.1). A cascade approach is a structure with a sequence of strong classifiers, and only those samples which pass all the these classifiers are considered to be positive.

The tree is trained recursively from the top node: At each tree node, the positive and negative training samples are gathered. Clearly, we cannot use all the available training data, which is in the order of 10^8. We always keep a subset of the training data which is passed from its parent node. But if the samples are too few, then bootstrapping is executed to pull more training data from the annotated dataset to the current node through the tree. The empirical distribution $\hat{q}(y)$ of the data is accumulated and we take an error measure by

$$e = (1 - \hat{q}(+1))(1 - \hat{q}(-1))$$

If either e is too small or the tree reaches its maximum depth on the current node, we stop the training at this node; otherwise, we train a strong boosting classifier on the samples. Once a strong classifier is learned, it divides all the training data at the current node into its left and right branches, which are then trained in the same way recursively.

We give the pseudocode for training a PBT as follows.
1) Given a set of volumes with a major sulcal curve annotated in each, randomly select 50 000 positive samples (a $15 \times 15 \times 15$ subvolume with the center voxel on the curve) and

50 000 negative samples (a $15 \times 15 \times 15$ subvolume with the center voxel on the background).
2) Compute the empirical distribution $\hat{q}(y)$ based on the weighted numbers of the positive and the negative samples; measure e.
3) If the training samples are too few, then perform bootstrapping to pull more training samples from the annotated dataset to the current node through the tree.
4) Check the stopping criteria: if either the current node reaches the maximum depth of the tree, or e is too small, stop the tree from expanding at this node.
5) Train a strong classifier at the node based on the current set of training samples using the standard boosting algorithm [26]. The candidate weak classifiers are classifiers defined on features such as curvatures, shape index, gradients, and 3-D Haar filters, etc.
6) Based on the strong classifier learned at the node, divide the training samples with soft weights, into its left and right branches, which are then trained in the same way recursively. In this paper, the threshold to pass the samples to the right branch is set as 0.6 and that to pass the samples to the left branch is 0.4. The samples with discriminative probabilities at the current node in between 0.4 and 0.6 are passed to both the branches.

More details of the training procedure of PBT can be found in [3]. Fig. 3 gives an illustration of the PBT.

B. Computing in PBT

In the testing procedure, the overall discriminative probability is computed as

$$\begin{aligned}
p(y|\mathbf{x}) &\approx \sum_{l_1} \hat{q}(y|l_1, \mathbf{x}) q(l_1|\mathbf{x}) \\
&\approx \sum_{l_1, l_2} \hat{q}(y|l_2, l_1, \mathbf{x}) q(l_2|l_1, \mathbf{x}) q(l_1|\mathbf{x}) \\
&\approx \sum_{l_1, \cdots, l_n} \hat{q}(y|l_n, \cdots, l_1) \cdots q(l_2|l_1, \mathbf{x}) q(l_1|\mathbf{x}) \quad (7)
\end{aligned}$$

where l_i's are augmented variables denoting the tree levels, as shown in Fig. 3. Here, we can also consider an augmented variable l_i as a hidden variable in the expectation maximization (EM) algorithm [27], though they are often not the same.

$l_i \in \{-1, +1\}$ indicates which branch is for this node: $l_i = -1$ and $l_i = +1$ point to the left and right branch, respectively. $q(l_i|l_{i-1}, \cdots, l_1, \mathbf{x})$ is the discriminative probability computed by the boosting strong classifier at the specified node according to (6). $\hat{q}(y|l_n, \cdots, l_1, \mathbf{x})$ is the empirical distribution at the leaf node. The testing process is performed in a divide-and-conquer manner, consistent with the learning process in PBT.

The following is the pseudo-code for computing (7) in PBT:
1) Given an input volume, we compute $p(y|\mathbf{x})$ for each voxel in the volume, i.e., $p(y(\mathbf{r})|\mathbf{V}(N(\mathbf{r})))$; where $\mathbf{x} = \mathbf{V}(N(\mathbf{r}))$.
2) Starting from the top node, compute the discriminative probability for the boosting strong classifier by (6), which corresponds to the term $q(l_i|l_{i-1}, \cdots, l_1, \mathbf{x})$ in (7).
3) If $q(l_i|l_{i-1}, \cdots, l_1, \mathbf{x})$ is big (e.g., >0.9), then pass the sample to the right branch of the node; if it is too small
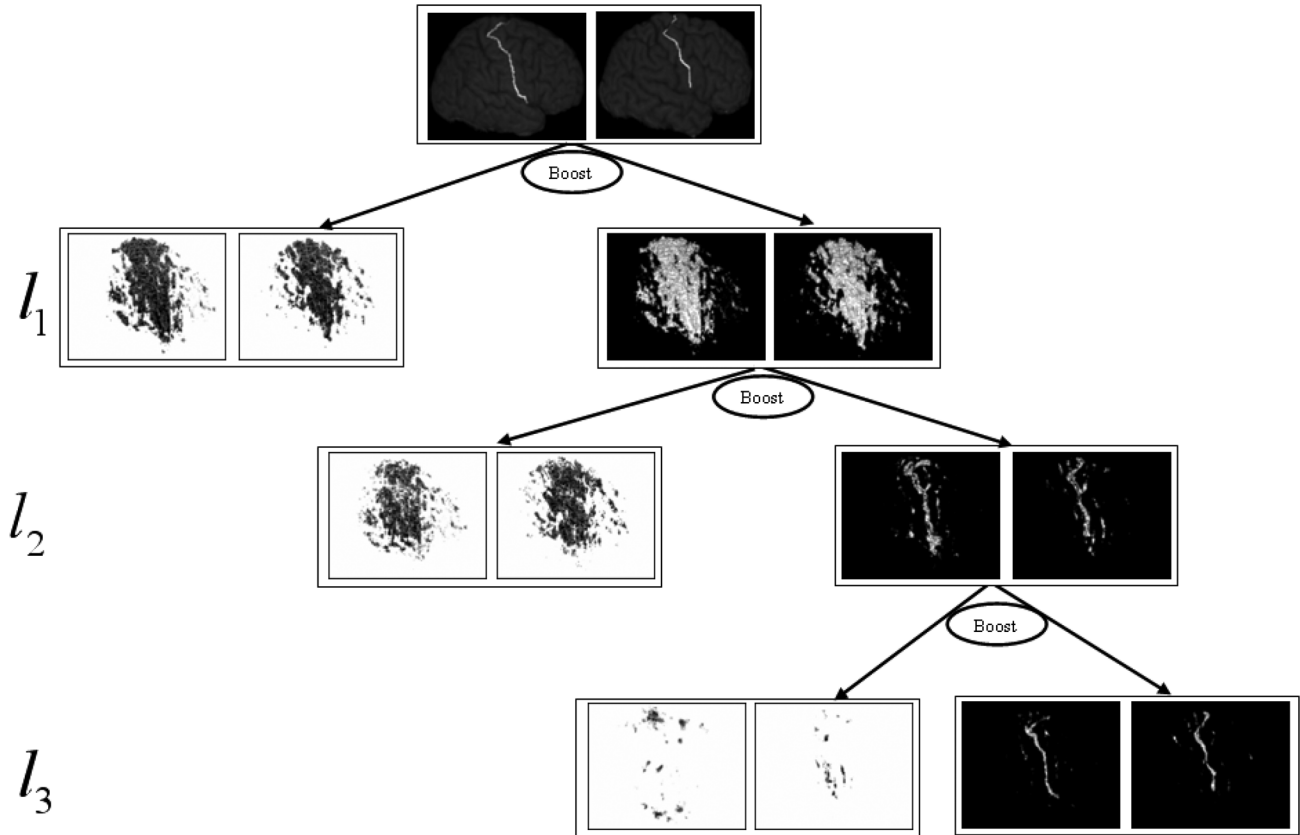
Fig. 3. Illustration of a boosting tree on two training volumes. The left branch shows the probability that each point is not on the sulcus curve while the right branch shows the probability that each point is on the sulcus curve.

(e.g., <0.1), then pass the sample to the left branch of the node; otherwise, pass the sample to both the branches.

4) Accumulate all the probabilities computed along the path according to (7).

The tree accumulates all the evidence from its descendants through a hierarchical structure.

As we can see from Fig. 3, the majority of the background voxels can be determined at the top levels of the tree with only a small amount of computation. Most of the effort is spent on the voxels which are ambiguous and they are resolved at the deep levels of the tree. Therefore, it is computationally efficient to compute the overall discriminative probability, even for all the voxels in the volume. It usually takes about 1 min to scan through the entire volume.

C. Features

The efficiency and effectiveness of the discriminative model trained by PBT are also largely determined by the choice of features/weak classifiers. Informative features help to resolve the ambiguities quickly and robustly. For example, previous studies [11] showed that curvature properties are useful for characterizing the sulcal curves. But no single feature can precisely characterize the entire sulcal pattern; moreover, in existing medical image approaches, there lacks an effective way to combine different features. Existing methods often rely on manually assigning weights to features, which requires difficult hand-tuning. By contrast, our goal is to use PBT to automatically select, combine, and weight different features in order to distinguish cortical sulcal curves from the background.

In the learning stage, we design a pool of approximately 8000 features for a subvolume of $15 \times 15 \times 15$ voxels, including intensity gradients, geometric features (e.g., principal curvatures, Gaussian curvatures, mean curvature, shape index, curvedness) [28], locations, and Haar filters. All these features except for the Haar filters are computed for the center voxel only, which is the voxel of interest. In this paper, these features are usually extracted at three scales to account for information across different scales. All the features including the Haars are computed in the subvolume. It is worth mentioning that voxel coordinates are more informative with better 3-D registration under which sulci will have higher chance to appear in some common places. Fig. 5 shows some typical types of Haar filters, which are then constructed at different sizes and computed at uniformly sampled locations in a $15 \times 15 \times 15$ subvolume. For each specific type of Haar filter with a specific size at a specific location, it computes a feature. If we use all possible Haar filters at all the locations, the number of possible features can be very large. Therefore, we only use a subset of locations which are regularly spaced in the subvolume. Each weak classifier is simply a stump classifier for each feature. It is essentially obtained by thresholding the feature response and the optimal value of the threshold is automatically decided by the boosting algorithm.

$$h\left(F_i(x)\right) = \begin{cases} +1 & if\ F_i(x) \geq th \\ -1 & \text{otherwise} \end{cases} \qquad (8)$$

where $F_i(x)$ is a feature computed on x and $h(F_i(x))$ is a weak classifier. The responses of 3-D Haar filters are rapidly computed using integral volumes [29]. At each location (x_1, y_1, z_1), the integral volume is computed as

$$\sum_{x=0}^{x_1-1} \sum_{y=0}^{y_1-1} \sum_{z=0}^{z_1-1} V(x, y, z)$$

where $V(x, y, z)$ is the intensity at location (x, y, z). Once the integral volume has been calculated, the responses of the Haar filters can be computed extremely fast. The intensity gradient features are computed directly and the location features are linear combinations of the voxel coordinates.

The calculations of geometric features are performed implicitly, i.e., we do not need to extract the surface, instead, we can calculate them based on the intensity values [28]. First we compute the first and second principal curvatures κ_1, κ_2 (with convention $\kappa_1 \geq \kappa_2$), then we calculate the mean curvature $(\kappa_1 + \kappa_2)/2$ and the shape index

$$SI = \frac{2}{\pi}\arctan\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}$$

(since $\arctan \in (-(\pi/2), \pi/2)$, then $SI \in (-1, 1)$). The shape index provides a continuous gradation between shapes, such as concave shapes $(-1 < SI < -1/2)$, hyperboloid shapes $(-1/2 < SI < 1/2)$, and convex shapes $(1/2 < SI < 1)$ [28]. The curvedness is defined to be $CV = \sqrt{(\kappa_1^2 + \kappa_2^2)/2}$.

Fig. 4 shows three features from the candidate pool which were selected by PBT. In the cortical valleys, the first principal curvatures have large values while the mean curvatures are somewhat smaller than the first principal curvatures; on the other hand, on the gyri, the first principal curvatures have relatively small values. Thus, the features address different aspects of the sulci and can be combined to cooperate and compensate for their individual limitations. The top 5 features selected by PBT for the first layer are all 3-D Haars, and we list the top three being: (0,0,0,13,13,9), (8,8,0,9,9,9), and (4,3,2,6,7,8). They are noted as (*left, top, front, right, bottom, back*) of the rectangle.

IV. THE ALGORITHM

In this section, we give the overall procedure of our cortical sulci detection algorithm, which works on both MRI volumes and cortical surfaces. We first discuss the problem of automatically detecting the starting and ending points.

A. End Point Detection

Detection of the starting and ending points (we refer to both as end points) is another challenging task in automatic sulci detection because hand-labeling them follows complex protocols [2]. The difficulty of detecting these end points prevents some of the existing algorithms [8] from being fully automatic. The determination of the end points in the literature is mostly based on rules about the geometric information, e.g., the points with highest curvature value [30].
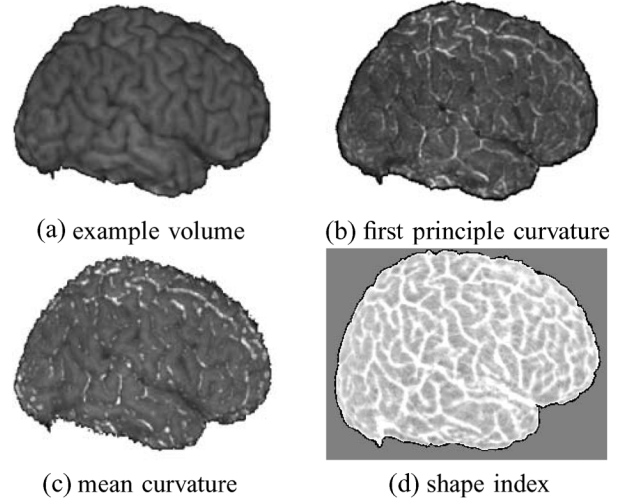

Fig. 4. Various curvature features: (a) an input volume; (b) the first principal curvature; (c) the mean curvature; (d) the shape index feature. These features tell somewhat different aspects about the sulci.
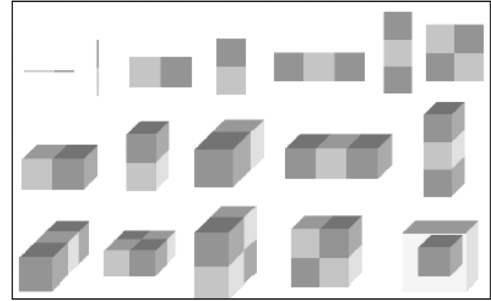

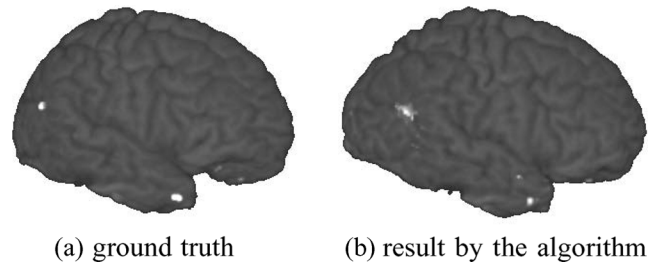Fig. 5. Different types of 1-D, 2-D, and 3-D Haar filters used.

Fig. 6. Illustration of the end points: (a) the two bright points show the end points of the Superior Temporal Sulcus labeled by a neuroanatomistm and (b) illustrates the probability of the end points by the algorithm.

In our preliminary work [17], we used a heuristic approach which performed reasonably well. We first used the training data to measure the mean and covariance of the positions of the end points, then we constrained the end points to lie within boxes centered on the means and with sides equal to twice the variance. We further localized the end points by requiring that $p(y(\mathbf{r}) = +1|\mathbf{V}(N(\mathbf{r}))) > T$ where T is a learned threshold. Of these remaining points, we selected the one that has the smallest value of $p(y(\mathbf{r}) = +1|\mathbf{V}(N(\mathbf{r})))$. This approach worked well on most images but occasionally it caused large errors.

Instead, we now use PBT to directly learn discriminative models for detecting the end points. The features, training, and

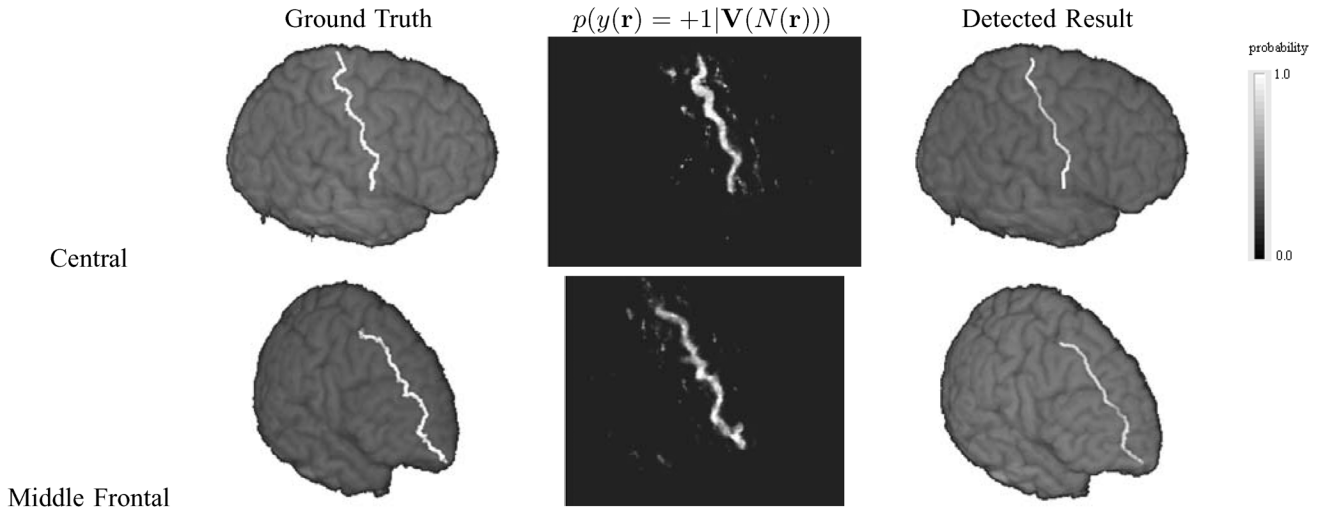| Sulcus	Training error by learning		Testing error by learning		Error by heuristics	
	Start Point (mm)	End Point (mm)	Start Point (mm)	End Point (mm)	Start Point (mm)	End Point (mm)
Central	1.43	2.14	4.60	7.36	7.46	11.18
Middle Frontal	2.88	3.88	4.00	5.80	6.93	13.43
Superior Temporal	2.28	1.82	3.12	7.32	9.07	15.90
Sylvian Fissure	2.83	1.82	4.24	9.44	9.76	15.22
Precentral	2.81	1.83	4.92	7.26	11.98	10.36

Fig. 7. Results for some of the MRI training images for detecting the Central Sulcus and the Middle Frontal Sulcus. First column shows the MRI volume with the ground truth superimposed. Second column displays the classification map $p(y(\mathbf{r}) = +1|\mathbf{V}(N(\mathbf{r})))$. Last column shows the detected sulcus in the MRI volumes.

testing proceed as before. The end points are detected as those with the highest discriminative probability.

In the experiments for detecting end points, we randomly select 15 volumes as training data, and the remaining 25 volumes as testing data. Fig. 6 shows the ground truth and the end points detected by our algorithm. For each detected point, an error is calculated as the Euclidean distance between the detected point and the ground truth. Table I shows the errors on the training data, the testing data, and the errors by the heuristic method. The learning approach has a clear advantage over our previous method [17].

B. Outline of the Algorithm

We give the outline of the algorithm below.

Training:
1) Train a discriminative model by PBT on the training volumes annotated with the major cortical sulci.
2) Train a discriminative model by PBT on the training volumes delineated with the end points of the major cortical sulcal curves.

Testing:
1) Given an input volume, compute the discriminative model $p(y(\mathbf{r}) = +1|\mathbf{V}(N(\mathbf{r})))$ for every voxel $\mathbf{r}$ in the volume.
2) Use the end point detector to detect the end points.

3) Use DP to detect a major cortical sulcal curve by minimizing the energy $E(C; \mathbf{V})$ in (1).

V. EXPERIMENTS

In this section, we show the experimental results by the proposed algorithm for detecting several major sulci, including the Central Sulcus, main body of the Superior Temporal Sulcus, the Postcentral Sulcus, the Middle Frontal Sulcus, and the Precentral Sulcus, which are shown in Fig. 1. We use a dataset consisting of 40 volumes [6], of which we randomly select 15 volumes for training and the remaining 25 volumes for testing. The ground truth is delineated by neuroanatomists on the surfaces extracted by the algorithm in [22]. We train and test our algorithm for data either as MRI volumes or extracted cortical surfaces (represented by the signal-distance function). The detected major sulci are compared to the ground truth and we obtain error measures for both the cases.

Three-dimensional MRI brain images were collected using a GE-Signa 1.5T scanner (General Electric, Milwaukee, MI). The same 3-D spoiled gradient echo pulse sequence was used for all participants—with the following parameters [6], echo time: 5 ms; repetition time: 24 ms; flip angle: 45°; number of excitations: 2 matrices, 256 × 192; field of view: 24 cm; slice thickness: 1.2 mm; 124 contiguous slices. We use resampled isotropic volumes with resolution 1 mm × 1 mm × 1 mm and the MRI volumes are 181 × 217 × 181 in size. The MRI images, cortical
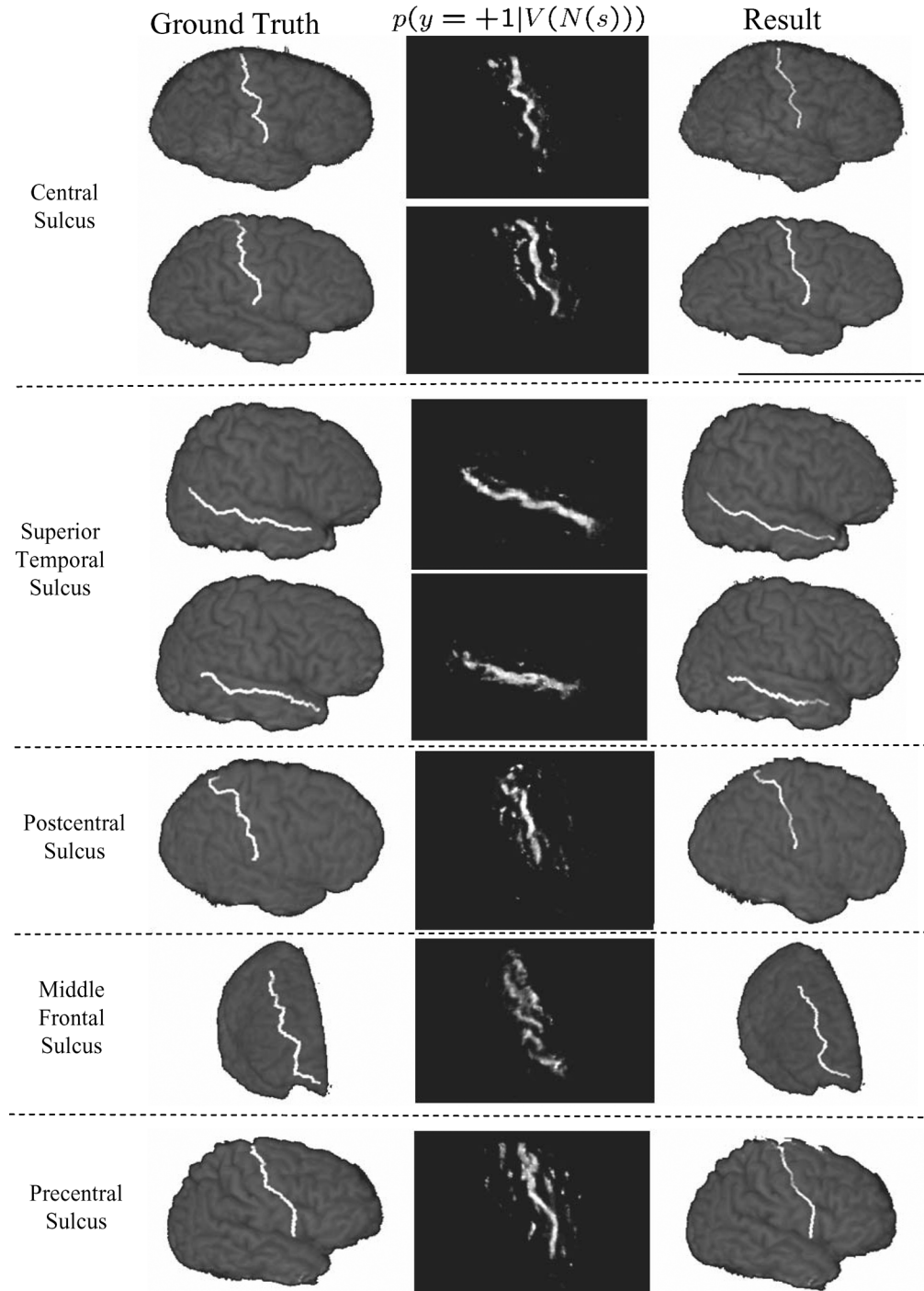
Fig. 8. Results on some of the MRI testing images for detecting the Central Sulcus, the Superior Temporal Sulcus, the Postcentral Sulcus, the Middle Frontal Sulcus, and the Precentral Sulcus. First column shows the MRI volume with the ground truth superimposed. Second column displays the classification map $p(y(\mathbf{r}) = +1|\mathbf{V}(N(\mathbf{r})))$. Last column shows the detected sulcus in the MRI volumes.

surfaces, and manually delineated sulcal lines are the same as in [6].

The computer used was an ordinary PC with 2.4 GHz CPU, and 1.0GB memory. Learning the PBT took approximately 8 h (it is a function of the size of the training dataset); computing the posterior probability for the PBT took approximately 1 min per image and running the dynamic programming took about 20 s per image. Standard code optimization techniques can reduce these times significantly.

A. Detection Results on Volume Data

We first train and test our algorithm on the MRI images. In these images, the length of the sulcus varies from 60–150 voxels. We perform experiments on several major sulcal curves including the Central Sulcus, the Precentral Sulcus, the main body of Superior Temporal Sulcus, the Middle Frontal Sulcus, the Sylvian Fissure, and the Postcentral Sulcus. For each type of major sulcus, we train a PBT classifier for the curve and a PBT

TABLE II
ERROR MEASURES ON 15 TRAINING AND 25 TESTING IMAGES FOR THE DETECTION OF SEVERAL MAJOR CORTICAL SULCI ON THE MRI IMAGES (SEE TEXT FOR THE NOTATION). UNIT OF DISTANCE IS MILLIMETERS

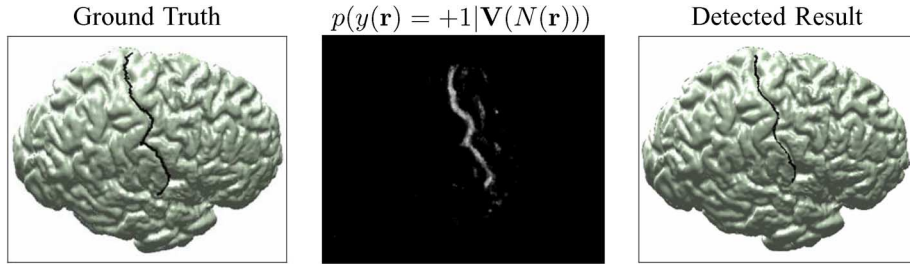| Sulcus		$< H_{av}(C,G) >$	$< H_{av}(G,C) >$	$< H_{wor}(C,G) >$	$< H_{wor}(G,C) >$
Central	Train	2.57	1.45	6.80	4.21
	Test	3.42	2.82	10.38	10.98
Superior Temporal (Main Body)	Train	3.83	2.91	10.34	8.03
	Test	4.17	3.56	11.26	10.19
Middle Frontal	Train	4.14	3.25	12.87	11.18
	Test	4.45	3.41	13.38	12.11
Sylvian Fissure	Train	2.91	1.84	8.44	6.47
	Test	3.50	3.48	10.26	11.90
Precentral	Train	3.66	2.67	11.42	9.70
	Test	4.68	3.64	13.10	10.88

Fig. 9. Results for one extracted surface for detecting the Central Sulcus. First column shows a surface with the ground truth superimposed. Second column displays the classification map $p(y(\mathbf{r}) = +1|\mathbf{V}(N(\mathbf{r})))$. Last column shows the detected sulcus on the extracted surface.

classifier for the end points. The maximum depth of the tree is fixed to be 9. Fig. 7 shows the detection results on two training volumes of the Central Sulcus and the Middle Frontal Sulcus. Fig. 8 shows some of the detection results on the testing data for different sulci. The first column shows the input volume with manual labels superimposed and the second column shows the discriminative probability maps. Note that the probability maps of the testing images have large responses around the correct position, but the maps are blurred, and sometimes disconnected. This implies that using the PBT classification model alone is not sufficient for extracting the sulcus, and so we need to use the information encoded in the shape prior. We then apply DP to extract the curve by minimizing (1) using the end points detected by PBT. The final results are shown in the third column of Figs. 7 and 8.

To quantitatively evaluate the performance of the proposed algorithm, we measure the distances between the detected major sulcal curves and their ground truths. We use the measures defined as

$$H_{av}(C,G) = \frac{1}{|C|} \sum_{c \in C} \min_{g \in G} |c - g|$$

$$H_{\text{wor}}(C,G) = \max_{c \in C} \min_{g \in G} |c - g|$$

where C denotes the detected curve and G is the ground truth annotated by neuroanatomists. These measures are variants of the Hausdorff distance, which is often used in the shape matching literatures [31]. Here $H_{av}(C,G)$ gives the average of the distances from curve C to their closest points on curve G. By contrast, $H_{\text{wor}}(C,G)$ measures the worst case fit from curve C to curve G. For symmetry, we also consider $H_{av}(G,C)$ and $H_{\text{wor}}(G,C)$. In the table, $\langle \cdot \rangle$ denotes the average over the dataset. For example,

$< H_{av}(C,G) > = (1/N) \sum_{i=1}^{N} H_{av}(C_i, G_i)$, where N is the number of examples in the dataset.

Table II shows the errors on the 15 training volumes and 25 testing volumes for several major cortical sulci. All these measures give values of H_{av} in the range 3–5 mm for both the testing and the training data. The testing errors are only slightly worse than the training errors, which suggests that our algorithm generalizes well. A reliability measure for six sulcal lines by different labelers was reported in [2] and the average differences for different sulci were mostly <2 mm.

B. Detection Results on Surface Data

We also applied our method directly to the cortical surfaces. The algorithm in [22] was first used to extract the surface from the input volume, then we computed the signal-distance function and used this as the input to the sulci extraction algorithm. We used the same set of features as before and the same PBT learning algorithm. Some features give no useful information when evaluated on the signal-distance function, and so PBT did not select them.

We repeated the same experiments to detect the Central Sulcus, the main body of Superior Temporal Sulcus, and the Sylvian Fissure. Fig. 9 shows the result on one training surface for detecting the Central Sulcus, in which the ground truth and the discriminative probability map are also displayed. Fig. 10 shows more results on the testing data for detecting the Central Sulcus, the Main Body of Superior Temporal Sulcus, and the Sylvian Fissure.

Table III evaluates the results of major sulci detected on the cortical surfaces, which are comparable to the results on the MRI volumes, see Table II. This indicates that our method has nearly the same performance on the two different types of input. Since the ground truth is delineated by neuroanatomists from the

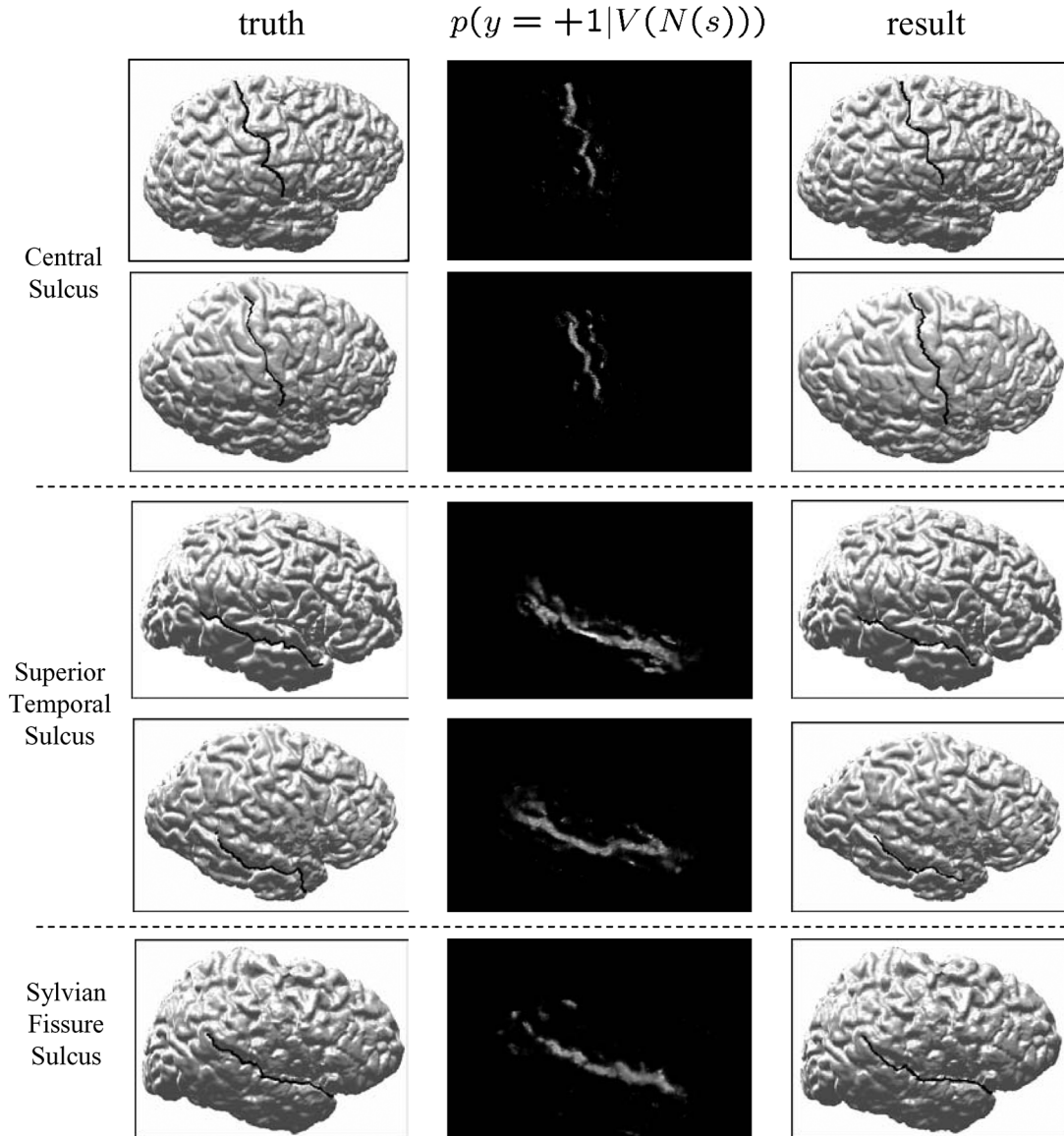truth $\qquad$ $p(y = +1|V(N(s)))$ $\qquad$ result

Fig. 10. Results on some of the cortical surfaces for detecting the Central Sulcus, the Superior Temporal Sulcus, and the Sylvian Fissure Sulcus. First column shows the cortical surfaces with the ground truth superimposed. Second column displays the classification map $p(y = +1|\mathbf{V}(N(\mathbf{r})))$. Last column shows the detected sulcus in the cortical surfaces.

surface data, we expect our algorithm to have better results on the MRI images than on the cortical surfaces, if the ground truth is annotated directly from the MRI images. This is because surfaces need to be extracted using a segmentation algorithm from the MRI volumes, and in general, introducing an intermediate stage increases the chance for errors.

Observe that although the worst case H_{wor} measures are large, the average distances H_{av} are small. This suggests that a few points can have large offsets, but the overall curves can still be detected accurately.

To see how our algorithm is influenced by the end point detection results, we also ran our algorithm to detect the central sulcus using manually annotated start and end points. The result is shown in Table IV.

Comparing Table IV with Table II, we see slight improvement for the average errors but more improvements for the worst situ-

ation. This suggests that large errors are due to end points which are inaccurately detected.

Table V shows a comparison between the results by Goualher *et al.* [4] and ours. We adopt their root mean square (rms) distance measures as

$$\mathrm{rms}^2 = 1\frac{1}{\mathrm{card}(\Omega)})\sum_{v \in \Omega} d(v)^2$$

were Ω is the set of all the voxels on a sulcus, $card(\Omega)$ is the number of the voxels, and $d(v)$ measures the distance between a voxel on a sulcus to the closest point on the sulcus to compare with. Both the mean rms and the maximum rms are shown in the table. Their results are slightly better than ours but our approach is fully automatic whereas theirs is semiautomatic. The sulci in our current training and testing dataset are manually la-

TABLE III
ERROR MEASURES ON 15 TRAINING AND 25 TESTING IMAGES FOR THE DETECTION OF SEVERAL MAJOR CORTICAL SULCI ON THE SURFACE DATA. (SEE TEXT FOR THE NOTATION). UNIT OF DISTANCE IS MILLIMETERS

Sulcus		$< H_{av}(C,G) >$	$< H_{av}(G,C) >$	$< H_{wor}(C,G) >$	$< H_{wor}(G,C) >$
Central	Train	2.61	1.72	10.41	8.63
	Test	3.06	2.27	10.41	10.40
Superior Temporal (Main Body)	Train	3.83	3.54	10.75	14.71
	Test	4.06	3.66	11.54	15.14
Sylvian Fissure	Train	3.14	2.69	9.52	12.78
	Test	3.40	2.74	10.85	14.1082

TABLE IV
ERROR MEASURES ON THE 25 TESTING IMAGES FOR THE DETECTION OF CENTRAL SULCUS ON THE VOLUME DATA USING MANUALLY ANNOTATED END AND START POINTS; THE RESULTS SHOULD BE COMPARED WITH THAT IN TABLE II

Central sulcus	$< H_{av}(C,G) >$	$< H_{av}(G,C) >$	$< H_{wor}(C,G) >$	$< H_{wor}(G,C) >$
Test	3.31	2.48	7.44	6.98

TABLE VI
ERROR MEASURES ON BRAINS WITH WILLIAMS SYNDROME FOR THE DETECTION OF SEVERAL MAJOR CORTICAL SULCI ON THE VOLUME AND THE SURFACE DATA (SEE TEXT FOR THE NOTATION). THE UNIT OF DISTANCE IS MILLIMETERS

Sulcus	$< H_{av}(C,G) >$	$< H_{av}(G,C) >$	$< H_{wor}(C,G) >$	$< H_{wor}(G,C) >$
Central (volume)	4.72	3.51	12.80	10.51
Central (surface)	3.18	2.32	11.50	11.64
Main Body of Superior Temporal (volume)	4.83	3.82	12.87	14.00
Main Body of Superior Temporal (surface)	4.09	3.77	12.54	15.14
Sylvian Fissure (volume)	4.45	3.80	11.56	12.90
Sylvian Fissure (surface)	3.59	3.77	11.29	15.30

TABLE V
COMPARISON BETWEEN THE RESULTS BY [4] AND OURS. THE UNIT OF DISTANCE IS MILLIMETERS

Central Sulcus	mean rms (auto vs. manual)	mean rms (manual vs. auto)
Goualher[4]	1.4	1.7
Ours on volumes	4.2	3.1
Ours on surfaces	3.4	3.2

beled on the brain surfaces segmented by the algorithm in [22], which may not always be very consistent with the original volumes. Better, more consistent, and more training data will likely further improve the quality of the results of our algorithm.

C. Experiments on Brains With Williams Syndrome

One of the reasons to detect major cortical sulci is to use them as landmarks, to assist the task of brain mapping and image registration, in order to study the abnormal patterns of diseased brains. We, therefore, further tested our algorithm on 40 brains with Williams Syndrome [6]. Table VI shows the error measures of several major sulcus curves. The tests were performed on both the volume data and the surface data. Compared with Tables II and III, we see that the performance on the diseased brains is slightly worse than that on the normal brains. Our algorithm can be directly used in [6] so that the efficiency of analyzing the brain diseases can be significantly improved. The manual annotation process took several weeks on this dataset while it took only about one minute for each major sulcus by our algorithm.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a learning based algorithm for automatic detection of major cortical sulci. The method combines a shape prior and a discriminative model, which is learned by PBT from an annotated database. A dynamic programming

method is used to detect the curves. The learning aspect of our algorithm plays a key role in this approach. The method is fully automatic and it is up to the learning algorithm to capture the complex folding patterns of the sulci from examples by automatically selecting and combining a set of features. The proposed algorithm is flexible and general, and has almost no parameters to tune for detecting different major sulci. Our algorithm works on both the MRI volumes and the extracted cortical surfaces. It is also fast and takes only about one minute to detect a major sulcus on a PC with 2.4 GHz CPU and 1.0 GB memory.

The method was applied to detect several major sulci (the Central Sulcus, the Superior Temporal Sulcus, the Postcentral Sulcus, the Middle Frontal Sulcus, and the Precentral Sulcus) and we gave detailed evaluations in this paper. We showed results on the training data, the testing data, and an additional set of Williams Syndrome brains. All these experiments were carried out both on MRI volumes and cortical surfaces. Furthermore, the performance on the Williams Syndrome cohort was similar to that on the normal brains. This demonstrates the adaptiveness of our algorithm.

In this work, we designed a feature pool consisting of about 8000 candidates such as curvatures, shape index, gradients, and 3-D Haar filters. Feature design is a critical issue in both medical imaging and computer vision problems. In the future, we plan to study the effectiveness of these features and design more informative features to further improve the efficiency and the robustness of our algorithm. We also plan to test our algorithm on a wider variety of brain images. We anticipate that our algorithm will help to improve the automation of brain mapping and registration. Our current approach detects each sulcal curve separately, and it will be interesting to see how to study the major sulci as a whole and learn their joint statistical regularities.

REFERENCES

[1] K. M. Hayashi, P. M. Thompson, M. S. Mega, C. I. Zoumalan, and S. S. Dittmer, Medial hemispheric surface gyral pattern delineation in 3-D: Surface curve protocol 2002 [Online]. Available: http://www.loni.ucla.edu/khayashi/Public/medial surface/,

[2] E. R. Sowell, P. M. Thompson, D. Rex, D. Kornsand, K. D. Tessner, T. L. Jernigan, and A. W. Toga, "Mapping sulcal pattern asymmetry and local cortical surface gray matter distribution in vivo: Maturation in perisylvian cortices," *Cereb. Cortex*, vol. 12, pp. 17–26, 2002.

[3] Z. Tu, "Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering," presented at the Int. Conf. Computer Vision, ICCV'2005, Beijing, China, Oct. 2005.

[4] G. Le Goualher, E. Procyk, D. L. Collins, R. Venugopal, C. Barillot, and A. C. Evans, "Automated extraction and variability analysis of sulcal neuroanatomy," *IEEE Trans. Med. Imag.*, vol. 18, no. 3, pp. 206–217, Mar. 1999.

[5] P. Hellier and C. Barillot, "Coupling dense and landmark-based approaches for nonrigid registration," *IEEE Trans. Med. Imag.*, vol. 22, no. 2, pp. 217–227, Feb. 2003.

[6] P. M. Thompson, A. D. Lee, R. A. Dutton, J. A. Geaga, K. M. Hayashi, 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," *Neuroscience*, vol. 25, no. 16, pp. 4146–4158, Apr. 2005.

[7] 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. L. Wang, T. G. M. van Erp, T. D. Cannon, and A. W. Toga, "Mapping cortical change in alzheimer's disease, brain development, and schizophrenia," *NeuroImage*, vol. 23, pp. 2–18, Sep. 2004.

[8] N. Khaneja, M. I. Miller, and U. Grenander, "Dynamic programming generation of curves on brain surfaces," *IEEE Trans. Patt. Anal. Mach. IntelL.*, vol. 20, no. 11, pp. 1260–1265, Nov. 1998.

[9] A. Bartesaghi and G. Sapiro, "A system for the generation of curves on 3-D brain images," *Human Brain Mapp.*, vol. 14, pp. 1–15, 2001.

[10] J. F. Mangin, D. Riviere, A. Cachia, E. Duchesnay, Y. Cointepas, D. Papadopoulos-Orfanos, P. Scifo, T. Ochiai, F. Brunelle, and J. Regis, "A framework to study the cortical folding patterns," *NeuroImage*, vol. 23, pp. 129–138, 2004.

[11] A. Caunce and C. J. Taylor, "Building 3-D sulcal models using local geometry," *Med. Image Anal.*, vol. 5, pp. 69–80, 2001.

[12] M. Ono, S. Kubik, and S. D. Abernathey, *Atlas of the Cerebral Sulci*. New York: Thieme Medical, 1990.

[13] X. Tao, J. L. 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, May 2002.

[14] D. Riviere, J. F. Mangin, D. Papadopoulos-Orfanos, J. M. Martinez, V. Frouin, and J. Regis, "Automatic recognition of cortical sulci of the human brain using a congregation of neural networks," *Med. Image Anal.*, vol. 6, pp. 77–92, 2002.

[15] 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, Sep. 1997.

[16] 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. 329–344, 2002.

[17] S. Zheng, Z. Tu, A. L. Yuille, A. L. Reiss, R. A. Dutton, A. D. Lee, A. M. Galaburda, P. M. Thompson, I. Dinov, and A. W. Toga, "A learning based algorithm for automatic extraction of the cortical sulci," in *Proc. 10th Int. Conf. Med. Image Comp. Comp. Assis. Interv.*, Copenhagen, Denmark, Oct. 2006, pp. 695–703.

[18] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu, "Statistical edge detection: Learning and evaluating edge cues," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 25, no. 1, pp. 57–74, Jan. 2003.

[19] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color and texture cues," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 26, no. 5, pp. 530–549, May 2004.

[20] P. Dollar, Z. Tu, and S. Belongie, "Supervised learning of edges and object contours," presented at the IEEE Conf. Comp. Vis. Patt. Recog., CVPR'2006, New York, Jun. 2006.

[21] D. Geman and B. Jedynak, "An active testing model for tracking roads from satellite images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 18, no. 1, pp. 1–14, Jan. 1996.

[22] J. D. MacDonald, "A method for identifying geometrically simple surfaces from three dimensional images," Ph.D. dissertation, McGill University, Montreal, QC, Canada, 1998.

[23] J. Besag, "Efficiency of pseudo-likelihood estimation for simple Gaussian fields," *Biometrika*, vol. 64, pp. 616–618, 1977.

[24] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comp. Vis.*, vol. 1, no. 4, pp. 321–333, 1987.

[25] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Nashua, NH: Athena Scientific, 2000.

[26] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Computer Sys. Sci.*, vol. 55, pp. 119–139, 1997.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Royal Statist. Soc. Series B*, vol. 39, pp. 1–38, 1977.

[28] H. Yoshida and J. Nappi, "Three-dimensional computer-aided diagnosis scheme for detection of colonic polyps," *IEEE Trans. Med. Imag.*, vol. 20, no. 12, pp. 1261–1274, Dec. 2001.

[29] Z. Tu, X. S. Zhou, D. Comaniciu, and L. Bogoni, "A learning based approach for 3-D segmentation and colon detagging," in *Euro. Conf. Comp. Vis., ECCV'2006*, Graz, Austria, May 2006, pp. 436–448.

[30] C. Renault, M. Desvignes, and M. Revenu, "3-D curves tracking and its application to cortical sulci detection," presented at the Int. Conf. Image Process., Vancouver, BC, Canada, Sep. 2000.

[31] R. Veltkamp and M. Hagedoorn, State-of-the-Art in Shape Matching Utrecht Univ., Utrecht, The Netherlands, Tech. Rep. UU-CS-1999-27, 1999.