Optimization Algorithms for Labeling Brain Sulci Based on Graph Matching

Faguo Yang and Frithjof Kruggel
Signal and Image Processing Lab, Department of Biomedical Engineering,
University of California, Irvine, CA 92697, USA
E-mails:{faguoy, fkruggel}@uci.edu

Abstract

Graph matching techniques are widely used in pattern recognition problems such as scene description, finger print identification, or face recognition. In this paper, we put forward two optimization methods for graph matching and compare them in the context of brain sulcus identification. The first approach is based on a constraint search in a neighborhood; the second uses a genetic algorithm for optimization. Experiments demonstrate that both methods yield satisfactory identification rates, however, the second method is more general and easier to adapt to similar problems.

1. Introduction

A symbolic description of an image is often represented as a graph, where vertices represent image regions (or their features) and edges between vertices correspond to relations between regions. To identify an object in an image, it is necessary to match a labeled template graph with a subject graph obtained from this specific image. Graph matching is one of the most complex problems in object recognition, pattern recognition and computer vision [1], because of its combinatorial nature. In the literature, genetic algorithms [2, 3], probability relaxation [4], expectation maximization [5, 6] are applied to solve this problem. No general method exists that solves all graph matching problems, so an optimization method has to be designed for a specific application. In this paper, we introduce and compare two graph matching optimization methods with their application in human brain sulcus identification. One is a neighborhood-constraint-search method (NS), the other is a genetic-algorithm-based optimization method (GA).

The neocortex of human brain is a convoluted sheet composed of gyri and sulci. In neurobiology, it is very useful to obtain an exact segmentation and identification of the sulci: to provide landmarks for 3D deformable brain image volume registration; to locate activation sites precisely in functional imaging; to describe morphological changes of brains affected by diseases.

Several pre-processing steps are required before building a graph representation: registration of 3D MRI brain images to the stereotaxic coordinate system [7], tissue segmentation [8], extraction of cerebral hemispheres, gray/white matter interface reconstruction [9] and segmentation of sulcal regions by a watershed-growing procedure [10, 11]. Then, sulcal segments can be represented as an attribute graph, where vertices contain the features of a sulcal segment and edges correspond to the neighborhood relations between the sulcal segments. We can identify brain sulci by matching a subject graph to a template graph representation that was constructed from a brain cortex segmentation labeled by an expert. Due to the high variability of the neocortical surface, we use multiple templates. So, we have to cope with a multi-model inexact graph matching problem. To solve our identification problem, we designed two optimization strategies and compare them in this paper.

Next, we introduce the problem of identifying brain sulci. Section 3 focuses on the neighborhood-constraint-search method; Section 4 is devoted to the genetic-algorithm-based optimization method. Experimental results for both approaches are compared in section 5. A discussion concludes the paper.

2. Human Sulcus Identification Problem

We now formally describe the formation of the graph representation, the segment features, and the objective function used in the sulcal recognition problem. Due to the limited space, pre-processing steps (brain segmentation, surface generation and surface parcellation into sulcal segments) are not discussed here but the subject of a separate publication [11].

2.1. Graph Representation of Sulcal Segments

From pre-processing, we obtained sulcal segments on the 3D surface mesh denoted by Q, where one or more segments correspond to a sulcus. Each segment on Q corresponds to a node V in the graph representation G. To define

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

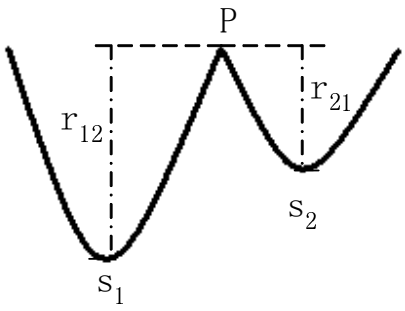

Figure 1. The ridge height between neighboring vertices of sulcal segments S_1 and S_2 is defined as the geodesic depth measured from the convex hull of the mesh.

neighborhood relationships, we dilate the segments on Q until no gyral vertices are left. The weight w between two nodes V_1, V_2 corresponding to two different sulcal segments S_1, S_2 is defined as:

$$w(V_1, V_2) = \exp(-(\alpha_1 d + \alpha_2 h)) \tag{1}$$

where d is the number of dilations required to remove all gyral vertices between the neighboring segments V_1, V_2; h is the average ridge height, which can be calculated by

$$h = \frac{1}{N_p} \sum_{i}^{N_p} (r_{12}^i + r_{21}^i)/2 \tag{2}$$

where N_p is the number of pairs of neighboring vertices between the two sulcal segments; The definition of r_{12} and r_{21} between a pair of neighboring vertices belonging to two different sulcal segments is illustrated in Fig. 1.

Smaller values of h and d yield a larger weight $w(V_1, V_2)$, and increase the probability that segments V_1, V_2 are merged to form a sulcus.

2.2. Features

We use position, orientation and shape as features that describe a sulcal segment, and compile them in a feature vector $\vec{f} = (f_1, f_2, \cdots, f_l)$. As the brain volumes are aligned with the stereotaxic coordinate system, orientation and position information are reliable for each sulcus. Each sulcus is found in a relatively constant region of the cortical surface, so location is critical for the identification task. We use the centroid c_x, c_y, c_z of each sulcal segment or combination of sulcal segments as location information. Positions are normalized by scaling the brain so that it fits into a bounding box of extent 1 in each direction. We use geometric moment invariants to describe the sulcus shape. There are thirteen rotation-invariant moments derived from the zero, second and third order moments. For detailed information, please refer to [12]. Sulcal orientation is an important feature for sulcus identification. We use the principal axes derived from the second order moments, denoted by d_x, d_y, d_z. A total of $l = 19$ features are used.

2.3. Objective Function

Let $s_1, s_2, \cdots, s_n$ denote the n sulci we want to recognize; $\psi_1, \psi_2, \cdots, \psi_n$ are the n sets of sulcal segments making up of candidates of sulci $s_1, s_2, \cdots, s_n$, respectively. A segment node V can only belong to a single sulcus: $\psi_i \bigcap \psi_j = \phi, \forall\, (i, j = 1, \cdots, n; i \neq j)$. The graph matching objective function can be defined as:

$$\begin{aligned} O(\psi_1, \psi_2, \cdots, \psi_n) &= O_F(\psi_1, \psi_2, \cdots, \psi_n) \\ &+ O_N(\psi_1, \psi_2, \cdots, \psi_n) \end{aligned} \tag{3}$$

where $O_F(\psi_1, \psi_2, \cdots, \psi_n)$ and $O_N(\psi_1, \psi_2, \cdots, \psi_n)$ are unary feature-based and neighborhood-based similarity values. The unary feature based similarity $O_F(\psi_1, \psi_2, \cdots, \psi_n)$ is defined by:

$$O_F(\psi_1, \psi_2, \cdots, \psi_n) = \sum_{i=1}^{n} S(\vec{f}_{\psi_i}, m_i), \tag{4}$$

where $\vec{f}_{\psi_i}$ is the feature vector of ψ_i; m_i is the sulcal feature model of sulcus s_i.

We assume each component of the feature vector $\vec{f}$ is Gaussian distributed, so we can define the similarity function $S(\cdot)$ as follows:

$$S(\vec{f}_{\psi}, m) = C_{\psi} \prod_i \frac{1}{\sqrt{2\pi}\sigma_i} \exp \frac{(f_i - \mu_i)^2}{2\sigma_i^2} \tag{5}$$

Where f_i is the ith component of feature vector $\vec{f}$; μ_i, σ_i are the mean value and standard deviation of feature component f_i for a specific sulcus, respectively; C_{ψ} is a coefficient related to the weights w_i between sulcal segments making up of candidate ψ, which is defined as:

$$C_{\psi} = \prod_{V \in \psi} u_V, \tag{6}$$

where u_V corresponds to the largest weight w that connects node V with a neighbor one level closer to the center node (See Fig. 2).

Larger values of $S(\vec{f}_{\psi}, m)$ indicate that segment set ψ is a better candidate for sulcus s characterized by model features m.

For the neighborhood-based similarity, we compare the neighborhood relationship among sulcal candidates $\psi_1, \psi_2, \cdots, \psi_n$ with that of the manually segmented sulci. The neighborhood relationships of the candidates $\psi_1, \psi_2, \cdots, \psi_n$ and the manually segmented sulci in the model represented by graph G_c and G_m, respectively, can

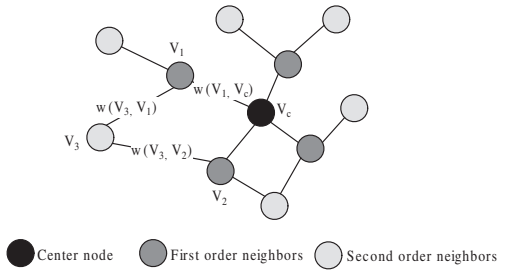

Figure 2. An illustration of the computation of the coefficient C_ψ. For the first order neighbor V_1 of the central node V_c, $u_{V_1} = w(V_1, V_c)$, because there are only one link between V_1 and the center node V_c; For the second order node V_3, there are two links connecting it to the first order neighbors V_1 and V_2, then we select the larger one of weights $w(V_3, V_1)$ and $w(V_3, V_2)$ as u_{V_3}.

be obtained from the graph representation G of the sulcal segments. There are three cases for the neighbors of a pair of matched nodes ψ and S from G_c and G_m (see Fig. 3). Sulcal candidates ψ, ψ_1, ψ_2 and ψ_3 are matched to s, s_1, s_2 and s_3, respectively. The first case is illustrated by ψ_1 and s_1, which are neighbors of the matched pair ψ and s. The second situation is demonstrated by ψ_2 and s_2, where ψ_2 is not a neighbor of ψ but s_2 is a neighbor of s. The third situation is illustrated by ψ_3 and s_3, where ψ_3 is a neighbor of ψ, while s_3 is not a neighbor of s. For each pair of matched nodes ψ and s from G_c and G_m, we compute the frequency of case one, case two and case three denoted by κ, η and γ, respectively. Then the neighborhood-based similarity can be defined as follows:

$$O_N(\psi_1, \psi_2, \cdots, \psi_n) = \beta \sum_{i=1}^{n} (\kappa_i - \eta_i - \gamma_i) \qquad (7)$$

where β is a weight for the neighborhood similarity. A higher similarity of the neighborhood in the model and template graph results in a higher similarity value O_N.

3. Neighborhood Constraint Search Method

In the first approach, we identify sulci sequentially and explicitly use the neighborhood information. Sulcal candidates for the pre- or postcentral sulcus are preselected as neighbors of the central sulcus. Thus, we only optimize Eq. 4, which does not include the neighborhood part.

Two steps are taken to solve the graph matching optimization problem here. First, we search a sulcal segment set candidate ψ_j^* for sulcus s_j that maximizes $S(\vec{f}_{\psi_j}, m_j)$, $j = 1, \cdots, n$. Then, if more than one sulcal set candidate ψ^* shares a specific segment, we assign it to a sulcus s that maximizes the objective function (Eq. 4).

We search the best sulcal segment set candidate ψ_j^* for sulcus s_j by examining 1 to N_s connected sulcal segments

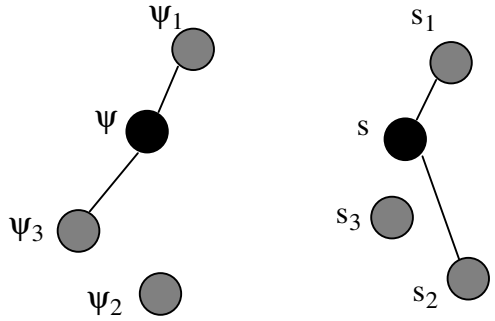

Figure 3. The three neighbor cases of a pair of matched nodes.

and recording the set ψ^* with a maximal similarity value. N_s cannot be too large because computation costs increase quickly; we use $N_s = 5$. To increase the chance that we found the best ψ^* in spite of a small N_s, we add a local search step: if including the neighbor node V_N of segment node $V \in \psi^*$ increases the similarity, then we add node V_N to ψ^*. Note this process continues until no new segment node is added to ψ^*. The algorithm is detailed below:

$n_s \leftarrow 1$
$\psi^* \leftarrow \phi$
$S(\psi^*, m) \leftarrow -\infty$
while n_s is less than N_s **do**
  for all node V in sulcal segment graph G **do**
    for all n_s neighboring-connected-segment set ψ with V as the center **do**
      Compute the similarity value $S(\psi, m)$ using formula (5)
      if $S(\psi, m)$ is greater than $S(\psi^*, m)$ **then**
        $S(\psi^*, m) \leftarrow S(\psi, m)$
        $\psi^* \leftarrow \psi$
      end if
    end for
  end for
  $n_s \leftarrow n_s + 1$
end while
for all sulcal segment node V in ψ^* **do**
  if add a neighbor V_N of V to ψ^* increase $S(\psi^*, m)$ **then**
    add V_N to ψ^*
  end if
end for

After the first step, we obtained segment candidates $\psi_1^*, \psi_2^*, \cdots, \psi_n^*$ for sulci $s_1, s_2, \cdots, s_n$, respectively. However, some of the sulcus candidates may share common segments, that is, not all $\psi_i^* \bigcap \psi_j^* = \phi$, $i \neq j$. We use the following method to solve this problem:

Find shared sulcal segment nodes V and identify their segment candidates ψ^*. Form a set ϑ of these segment

candidates and store the sets in a list L.
while end of list L has not been reached **do**
Select the current shared sulcal segment node V and its corresponding ϑ as the next items in list L.
for all $\psi^* \in \vartheta$ **do**
assign V into ψ^*.
exclude V from all other elements of ϑ.
compute the objective function $O(\vartheta)$ using formula (4).
end for
assign sulcal segment V into a sulcal segment candidate with maximized $O(\vartheta)$.
end while

Now, all nodes V representing sulcal segments are uniquely addressed to segment candidates ψ^*, and are recognized as a specific sulcus.

4. Genetic Algorithm Based Optimization Method

In the second strategy, we try to recognize all sulci simultaneously. We make use of the neighborhood information implicitly by maximizing Eq. 3, and optimize the objective function by a genetic algorithm.

4.1. Overview of Genetic Algorithms

Genetic algorithms [13] are effective to find the global optimum by introducing heuristic techniques for combinatorial problems [14]. They try to mimic the evolution process in the biological field. A solution that fits the objective function better has a higher chance to survive and produce offsprings. It is necessary to define the following to solve an optimization problem using genetic algorithms.

Representation: A representation in genetic algorithms is a form of expression of the solution of the optimization problem. We should be able to represent any solution to our problem and if possible, do not represent any infeasible solution. If infeasible solutions are possible, then the objective function must be designed to penalize them. An optimization parameter of a specific problem is represented as a *gene*, the set of all parameters - a solution - is called *chromosome*, a group of chromosomes forms a *population*.

Genetic operators: The basic genetic operators include initialization, mutation, and crossover. The initialization operator defines how to create initial solutions of a specific problem. The crossover operator uses two or one existing solutions to produce two or one new solutions. The mutation operator tells how to vary some parts of a solution to generate a new solution.

Fitness function: A fitness function is used to calculate the quality of a solution. The fitness function is related to the objective function or can be the objective function itself.

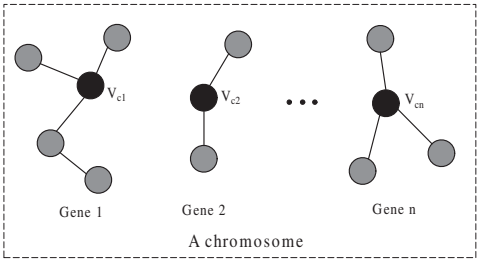

Figure 4. In the second optimization approach, each sulcus is represented by a gene and corresponds to a set of candidate segments.

With the above concepts and notations, the basic steps of the Genetic Algorithm are as follows:

Step 1: Use an initialization operator to initialize the population.

Step 2: Select some chromosomes according to a *crossover probability* to produce a group of new chromosomes.

Step 3: Mutate the offsprings produced by the crossover operator based on a *mutation probability*.

Step 4: Insert the newly generated chromosomes into the population. Delete some chromosomes with a low fitness to keep the population at a constant number of chromosomes.

step 5: If stop criteria are not yet satisfied, go to *step 2*, else stop.

4.2. A Genetic Algorithm for Sulcus Recognition

Remember that a sulcus in our recognition problem consists of one or more connected segments. We try to label n sulci by finding n best matching sets of connected segments. Each set is represented by a gene. Because gene g_i in a chromosome corresponds to a candidate ψ_i for sulcus s_i, genes must accomodate different numbers of segment nodes, which is guaranteed by a specific design of the initialization, crossover and mutation operators. A chromosome is illustrated in Fig. 4.

The representation affects the dimensionality of the search space and, thus, the convergence speed of the algorithm. A straightforward alternative representation of our problem is a bit string. If we try to recognize 7 sulci, we can use 3 bits to represent the label of each sulcal segment. The label "0" denotes segments not addressed to any of the 7 sulci. Suppose there are N_G segment nodes in the graph, so the total string length will be $3\,N_G$. Therefore

Table 1. Sulcus correct recognition rate in 100 test subjects for the genetic algorithm (GA), and neighborhood search (NS). CS: Central sulcus; STS: Superior temporal sulcus; INS: Insula; PCS: Precentral sulcus; PTS: Postcentral sulcus; CIS: Cingulate sulcus; CLS: Callosal sulcus.

	CS	STS	INS	PCS	PTS	CIS	CLS
GA	87%	86%	96%	83%	78%	91%	95%
NS	92%	82%	87%	82%	88%	83%	86%

the search space has $2^{3\,N_G}$ solutions. For the first method, we have a much smaller upper limit of $C_{N_G}^{7} C_{N_G}^{N_{max}} N_{max}$ where N_{max} is the maximum number of segment nodes each gene can have.

The initialization operator is utilized to generate N_c chromosomes to form the initial population. For each gene g in a chromosome, we randomly select a center node V_c and put it into g. Then we randomly select a neighbor node of those nodes in g, which is not already included in the chromosome, and add it to g until we have N_s segment nodes in g. N_s is a randomly selected number between $1, \cdots, N_{max}$. This initialization method ensures that different genes in a chromosome have different segment nodes and the segment nodes in each gene are connected.

The crossover operator defines how to generate new solutions from existing solutions. We randomly select a number k from $1, \cdots, n$, then we exchange the two genes g_k from the two existing chromosomes Pa_1 and Pa_2. If the offsprings violate the rule that different genes in a chromosome cannot have same segment nodes, we randomly select and exchange another pair of genes until we obtain two correct offsprings. In the case that we tried all n genes, and still cannot obtain two correct offsprings, we view one or two of the existing parent chromosomes as the offsprings.

The mutation operator modifies newly generated chromosomes with a specific *mutation probability*. This operator randomly selects a number k from $1, \cdots, n$, and generates a new gene using the gene-generation method in the initialization operator to replace gene g_k. When creating the new gene g_k, we should not use those segment nodes already included in g_i, $(i = 1, \cdots, k-1, k+1, \cdots, n)$ of the same chromosomes.

We terminate the genetic algorithm if the fitness values of the best chromosome and the worst chromosome of the current generation are equal or there are over N_p generations ($N_p = 50$ here).

To speed up convergence, we introduce a local search step near the segment nodes of the best chromosome of each generation, which will be described in the next sub-section.

The second approach can be described as follows:

Step 1: Use the initialization operator to produce N_c chromosomes to form the population.

Step 2: Choose $N_c\, R_p$ chromosomes from the current population to form a temporary population, where R_p is the chromosome replacement ratio. The selector picks a chromosome based on its fitness score relative to the rest of the population. Any chromosome can be chosen by the probability equal to the fitness of the chromosome divided by the sum of the fitness of the chromosomes in the population. Then randomly select $N_c R_p P_c/2$ pairs of chromosomes from the temporary population, where P_c is the crossover probability. Finally, the crossover operator generates $N_c R_c P_c$ offsprings and replaces their parents in the temporary population.

Step 3: Randomly select $N_c R_p P_m$ chromosomes from the temporary population and use the mutation operator to modify them.

Step 4: Add $N_c R_p$ chromosomes from the temporary population to the current population, then remove the $N_c R_p$ worst chromosomes to form the next population.

Step 5: Search near the best chromosome of the population using the method described in next subsection. If the newly searched chromosome has a higher fitness than the current best chromosome, replace it.

Step 6: If the termination condition is satisfied, stop; Or go to the step 2.

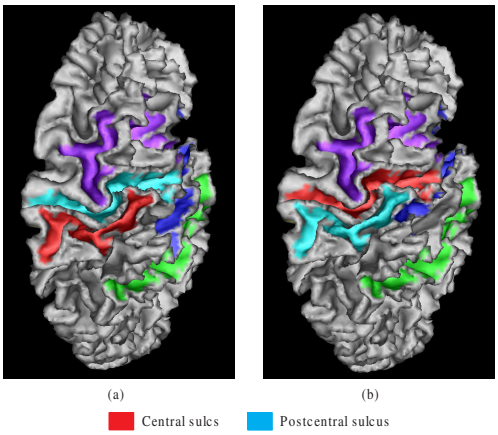

Figure 5. In this case, NS correctly identified the CS (see (b)), while GA failed (see (a))

.

4.3. Local Search

Performing a local search around the current best solution can speed up the optimization process. The local search employs a two-step strategy. First, we search a sulcal segment set candidate ψ_j^* (gene g_j^*) for sulcus s_j that maximizes $S(\vec{f}_{\psi_j}, m_j)$, $j = 1, \cdots, n$, in the neighborhood of

the segment nodes of gene g_j of the best chromosome in each generation. Second, if more than one sulcal set candidate ψ^* shares a specific segment, we assign it to a sulcus s that maximizes the objective function (Eq. 4). The local search method here is similar to the search method in section 3. Here, we just search around the single best solution of each generation while in the first method, all segment nodes in the subject graph were considered.

5. Experimental Results

We used 120 MR T_1-weighted image datasets of healthy subjects [7] to evaluate our method. Surfaces representing the GM/WM interface were computed and segmented into sulcal substructures. In a training set of 20 cases, a neuro-anatomist manually labeled the central sulcus (CS), precentral sulcus (PCS), postcentral sulcus (PTS), insula (INS), superior temporal sulcus (STS), cingulate sulcus (CIS), and callosal sulcus (CLS).

Feature vectors were computed for all sulci of all training cases, and the mean and standard deviation of each feature component were used for recognizing sulci of new cases. The remaining 100 MRI brain datasets were processed by both methods using the parameter settings from the training set. The neuro-anatomist rated the automatic identification: a result is considered as correct if all but only those segments are correctly labeled that make up a specific sulcus, even if it is interrupted.

We chose $\alpha_1 = 2$, $\alpha_2 = 10.5$ in Eq. 1 for both optimization methods. The parameters in the genetic algorithm producing the best average recognition rate were: $N_c = 2000$, $R_p = 0.4$, $P_c = 0.9$, $P_m = 0.2$, $N_{max} = 10$. The identification rates are displayed in Tab. 1. The average identification rates are 88% for the genetic algorithm (GA) and 85.7% for the neighborhood search (NS). There is a general and consistent advantage of GA over NS. The slight advantage of NS for CS and PTS is explained by the heuristic implementation of the search strategy: first, identify CS, then PCS, then PTS in the neighborhood. GA tries to recognize all sulci simultaneously, with a certain chance of mixing them up (see Fig. 5). The neighborhood search method correctly recognized the central sulcus (see Fig. 5(b)), while the GA optimization method wrongly identified the central sulcus as the post-central sulcus (see Fig. 5(a)).

To provide a visual impression of the identification results, we show results of the genetic-algorithm-based optimization method for three cases in Fig. 6. All 7 primary sulci (CS, PCS, PTS, STS, CIS, CLS, INS) are correctly identified, though there is a considerable variability of the corresponding sulci.

The algorithms were implemented in the C++ language. The computation time is 60 min for GA and 10 min for NS, on a Linux server with a 2.21 GHz AMD Athlon 64 bit processor and 4 GB memory.

6. Discussion

We introduced two graph matching strategies for an automatical labeling of brain sulci. Both achieve satisfactory identification rates. The NS method features a lower computational demand and high identification rates for CS and PTS. In contrast, GA naturally takes neighborhood information into account. It is easier to generalize for identifying additional sulci and has a better overall recognition rate. Probability relaxation [4] can be considered as another optimization method and will be investigated in the future.

References

[1] E. Bienenstock and C. von der Malsburg. A neural network for invariant pattern recognition. Europhysics Letters, 4(1):121-126, 1987.

[2] K. G. Khoo and P. N. Suganthan. Evaluation of genetic operators and solution representations for shape recognition by genetic algorithms. Pattern Recognition Letters, 23(13):1589-1597, 2002.

[3] A. D. J. Cross, R. Myers, and E. R. Hancock. Convergence of a hill-climbing genetic algorithm for graph matching. Pattern Recognition, 33(11):1863-1880, 2000.

[4] W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. IEEE Trans. Pattern Analysis and Machine Intelligence, 17(8):749-764, 1995.

[5] B. Luo, and E. R. Hancock. Structural graph matching using the EM algorithm and singular value decomposition. IEEE Trans. Pattern Analysis and Machine Intelligence, 23(10):1120-1136, 2001.

[6] A. D. J. Cross and E. R. Hancock. Graph matching with a dual-step EM algorithm. IEEE Trans. Pattern Analysis and Machine Intelligence, 20(11):1236-1253, 1998.

[7] F. Kruggel and D. Y. von Cramon. Alignment of magnetic-resonance brain datasets with the stereotactical coordinate system. Medical Image Analysis, 3(2):175-185, 1999.

[8] D. L. Pham and J. L. Prince. Adaptive fuzzy segmentation of magnetic resonance images. IEEE Trans. Medical Imaging, 18(9):737-752, 1999.

[9] F. Kruggel. Techniques in analyzing the neocortial fine-structure. In: Leondes C.T. (ed.), Medical Imaging Systems Vol. 5. World Scientific Press, Singapore, 2005.

[10] F. Meyer and S. Beucher. Morphological segmentation. Journal of Visual Communication and Image Representation, 1:21-46, 1990.

[11] F. Yang and F. Kruggel. Automatic segmentation of human brain sulci. Submitted to Medical Image Analysis.

[12] C.-H. Lo and H.-S. Don. 3-D moment forms: their construction and application to object identification and positioning. IEEE Trans. Pattern Analysis and Machine Intelligence, 11(10): 1053-1064.

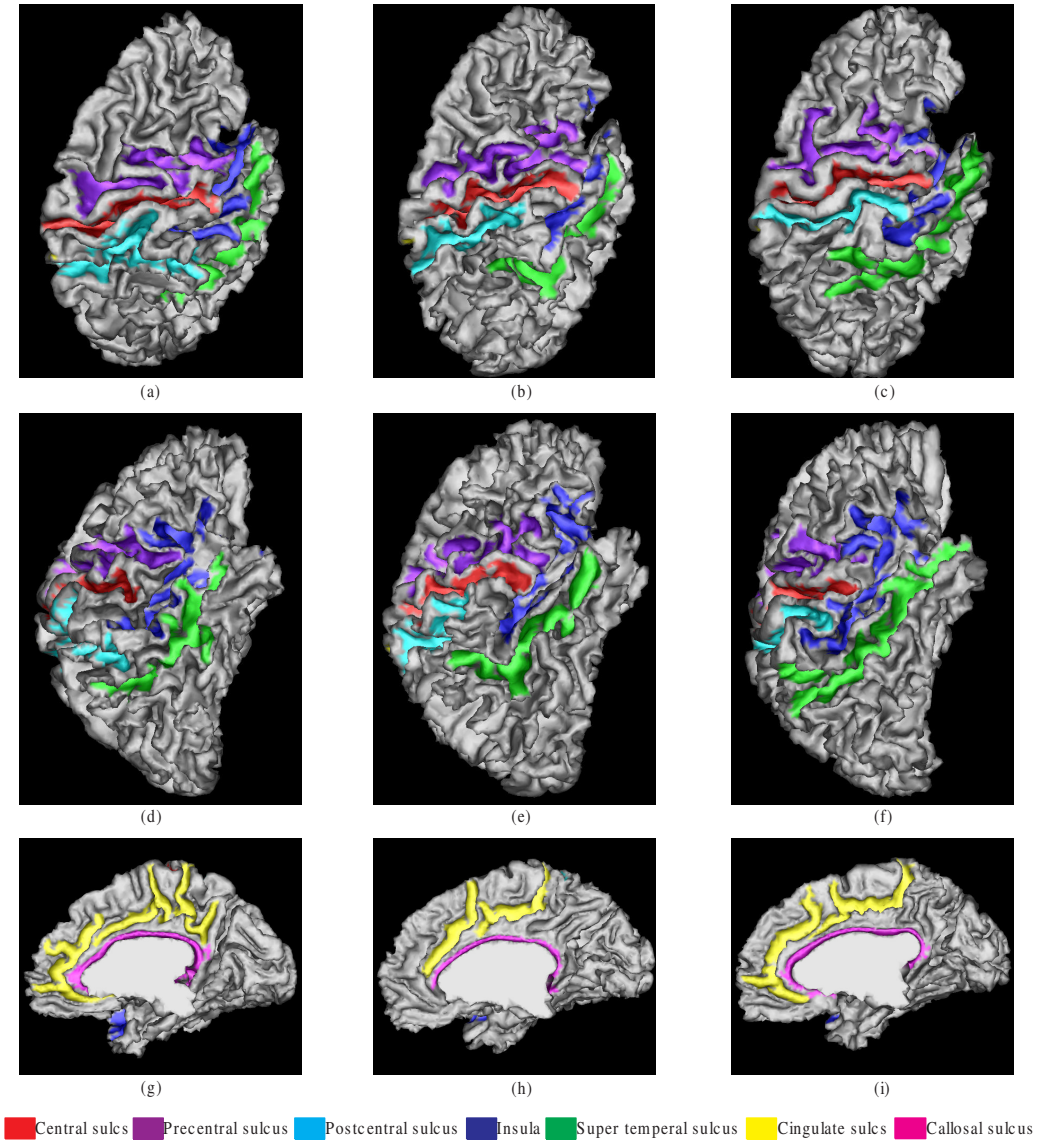

Figure 6. Identification results for three subjects.

[13] D. E. Goldberg. Genetic algorithm in search, optimization and machine learning. Reading, MA: Addison-Wesley, 1989.

[14] F. Yang and T. Jiang. Cell image segmentation with kernel-based dynamic clustering and an ellipsoidal cell shape model. Journal of Biomedical Informatics, 34:67-73, 2001.